



Завод за унапређивање
образовања и васпитања



Центар за стручно образовање
и образовање одраслих



МАТУРСКИ ИСПИТ ЕЛЕКТРОТЕХНИЧАР ИНФОРМАЦИОНИХ ТЕХНОЛОГИЈА - ОГЛЕД

Приручник о полагању матурског испита
у образовном профилу електротехничар информационих
технологија - оглед

Ревизија:
Београд, јануар 2021.

САДРЖАЈ

САДРЖАЈ.....	1
УВОД	4
НОВИ КОНЦЕПТ МАТУРСКОГ ИСПИТА	5
ПРОГРАМ МАТУРСКОГ ИСПИТА.....	6
1. ЦИЉ МАТУРСКОГ ИСПИТА.....	6
2. СТРУКТУРА МАТУРСКОГ ИСПИТА.....	6
3. ОРГАНИЗАЦИЈА МАТУРСКОГ ИСПИТА.....	6
4. УСЛОВИ ЗА ПОЛАГАЊЕ МАТУРСКОГ ИСПИТА	6
5. ЕВИДЕНЦИЈА НА МАТУРСКОМ ИСПИТУ	7
6. ДИПЛОМА И УВЕРЕЊЕ	7
7. ИСПИТ ИЗ МАТЕРЊЕГ ЈЕЗИКА И КЊИЖЕВНОСТИ	7
7.1 СТРУКТУРА ИСПИТА	8
7.2 ОЦЕЊИВАЊЕ	8
7.3 ОРГАНИЗАЦИЈА ИСПИТА	8
8. ИСПИТ ЗА ПРОВЕРУ СТРУЧНО–ТЕОРИЈСКИХ ЗНАЊА	8
8.1 СТРУКТУРА ИСПИТА	8
8.2 ОЦЕЊИВАЊЕ	9
8.3 ОРГАНИЗАЦИЈА ИСПИТА	9
9. МАТУРСКИ ПРАКТИЧНИ РАД.....	10
9.1 СТРУКТУРА ИСПИТА	10
9.2 ОЦЕЊИВАЊЕ	14
9.3 ОРГАНИЗАЦИЈА ИСПИТА	14
АНЕКС 1 ЗБИРКА ТЕОРИЈСКИХ ЗАДАТКА За МАТУРСКИ ИСПИТ	16
ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК С	18
ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК С#.....	62
ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК ЈАВА	94
ВЕБ ДИЗАЈН	126
ВЕБ ПРОГРАМИРАЊЕ	133
ИНФОРМАЦИОНИ СИСТЕМИ И БАЗЕ ПОДАТКА	138
АНЕКС 2 РАДНИ ЗАДАЦИ СА ОБРАСЦИМА ЗА ОЦЕЊИВАЊЕ НА МАТУРСКОМ ПРАКТИЧНОМ РАДУ....	171
КОМПЕТЕНЦИЈА А.....	173
ЕИТ-А2: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ФОТО ГАЛЕРИЈА.....	176
ЕИТ-А3: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА	179
ЕИТ-А4: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СРБИЈА И ЊЕНИ СУСЕДИ.....	182
ЕИТ-А5: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕЗЕРВАЦИЈА АУТОБУСКИХ КАРАТА	185
ЕИТ-А6: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТЕЛЕФОНСКИ ИМЕНИК	188
ЕИТ-А7: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕД ВОЖЊЕ	191
ЕИТ-А8: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВЕБ ПРОДАВНИЦА	194
ЕИТ-А9: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КОНВЕРТОР ТЕКСТА.....	197

ЕИТ-А10: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КУРСНА ЛИСТА	200
ЕИТ-А11: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – НУМЕРОЛОШКО СЛАГАЊЕ	203
ЕИТ-А12: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КЊИГА УТИСАКА	206
ЕИТ-А13: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЦД КАТАЛОГ	209
ЕИТ-А14: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТВ ПРОГРАМ	212
ЕИТ-А15: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВОЗНИ ПАРК.....	215
ЕИТ-А16: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РАСПОРЕД ЧАСОВА	218
ЕИТ-А17: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ ТЕСТ.....	221
ЕИТ-А18: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА	224
ЕИТ-А19: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ РЕЧНИК	227
ЕИТ-А20: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СЛАЈДЕР ЗА СЛИКЕ (Image slider)	230
ЕИТ-А21: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СОРТИРАЊЕ ТАБЕЛЕ	233
ЕИТ-А22: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕФЕКТИ ТАБЕЛЕ	236
ЕИТ-А23: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КАРТИЦЕ ЛОЈАЛНОСТИ	239
ЕИТ-А24: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – УСПЕХ УЧЕНИКА	242
ЕИТ-А25: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – БИБЛИОТЕКА	245
ЕИТ-А26: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЗВУЦИ ИНСТРУМЕНТА.....	248
ЕИТ-А27: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – БИОРИТАМ	251
КОМПЕТЕНЦИЈА Б.....	254
ЕИТ-Б1: ИЗЛОЖБЕ ПАСА	254
ЕИТ-Б2: ИЗЛОЖБЕ ПАСА	258
ЕИТ-Б3: ДВД КОЛЕКЦИЈА.....	262
ЕИТ-Б4: ДВД КОЛЕКЦИЈА.....	266
ЕИТ-Б5: ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ	270
ЕИТ-Б6: ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ	274
ЕИТ-Б7: ТУРИСТИЧКА АГЕНЦИЈА.....	278
ЕИТ-Б8: ТУРИСТИЧКА АГЕНЦИЈА.....	282
ЕИТ-Б9: АНТИКВИТЕТИ И ЛОКАЦИЈЕ	286
ЕИТ-Б10: АНТИКВИТЕТИ И ЛОКАЦИЈЕ	290
ЕИТ-Б11: ПОЗОРИШНЕ ПРЕДСТАВЕ.....	294
ЕИТ-Б12: ПОЗОРИШНЕ ПРЕДСТАВЕ.....	298
ЕИТ-Б13: КОЛЕКЦИЈА КЊИГА	302
ЕИТ-Б14: КОЛЕКЦИЈА КЊИГА	306
ЕИТ-Б15: ШКОЛСКА БИБЛИОТЕКА	310
ЕИТ-Б16: ШКОЛСКА БИБЛИОТЕКА	314
ЕИТ-Б17: ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА.....	318
ЕИТ-Б18: ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА.....	322
ЕИТ-Б19: ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ	326
ЕИТ-Б20: ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ	330
ЕИТ-Б21: ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА.....	334
ЕИТ-Б22: ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА.....	338
ЕИТ-Б23: ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ.....	342

ЕИТ-Б24: ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ.....	346
ЕИТ-Б25: КОМУНИКАЦИЈА СТУДЕНТА И ПРОФЕСОРА	350
ЕИТ-Б26: КОМУНИКАЦИЈА СТУДЕНТА И ПРОФЕСОРА	354
ЕИТ-Б27: ДНЕВНИ БОРАВАК ДЕЦЕ	358
ЕИТ-Б28: ДНЕВНИ БОРАВАК ДЕЦЕ	362
ЕИТ-Б29: РЕЗЕРВАЦИЈА ХОТЕЛСКИХ СОБА	366
ЕИТ-Б30: РЕЗЕРВАЦИЈА ХОТЕЛСКИХ СОБА	370
ЕИТ-Б31: АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА	374
ЕИТ-Б32: АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА	378
ЕИТ-Б33: ГОЛФ КЛУБ	382
ЕИТ-Б34: ГОЛФ КЛУБ	386
ЕИТ-Б35: ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ	390
ЕИТ-Б36: ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ	394
ЕИТ-Б37: СЕОСКИ ТУРИЗАМ	398
ЕИТ-Б38: СЕОСКИ ТУРИЗАМ	402

УВОД

Модернизација друштва и усмереност ка економском и технолошком развоју подразумевају иновирање како општих, тако и специфичних циљева стручног образовања. У том смислу стручно образовање у Србији се, пре свега, мора усмеравати ка стицању стручних компетенција и постизању општих исхода образовања, неопходних за успешан рад, даље учење и постизање веће флексибилности у савладавању променљивих захтева света рада и друштва у целини и већу мобилност радне снаге.

Да би се обезбедило побољшање квалитета, укључиле интересне групе и социјални партнери, обезбедио ефикасан трансфер знања и стицање вештина код свих учесника у образовном процесу уз пуно уважавање етничких, културолошких и лингвистичких различитости, Министарство просвете, науке и технолошког развоја Републике Србије започело је припреме за реорганизацију и реформу система стручног образовања, доношењем Стратегије развоја стручног образовања у Србији¹ коју је усвојила Влада Републике Србије децембра 2006. године, акционог плана² за њено спровођење, усвојеног марта 2009. године и Стратегије развоја образовања у Србији до 2020. године³ усвојене новембра 2012. године.

У подручју рада *Електротехника и рачунарство од школске 2012/13. године*, уведен је нови огледни наставни програм: електротехничар информационих технологија - оглед⁴. Овај програм развијан је на основу анализе и описа занимања за које се ученици школују у овом образовном профилу. Провера стечености програмом прописаних знања, вештина и главних стручних компетенција подразумева увођење новог концепта матурског испита.

Прва генерација ученика образовног профиле електротехничар информационих технологија—оглед, завршава своје школовање полагањем матурског испита школске 2015/16. Године.

Програм матурског испита у свим огледним образовним профилима припремљен је уз консултације и према захтевима социјалних партнера – Уније послодаваца, Привредне коморе Србије, одговарајућих пословних удружења и уз активно учешће средњих стручних школа у којима се оглед спроводи. Овај програм настао је на основу свеобухватног истраживања различитих међународних концепата матурског испита, уз уважавање постојећих искустава у овој области у Републици Србији.

Овај приручник је јавни документ, намењен ученицима и наставницима средњих стручних школа у којима се спроводи огледни наставни програм електротехничар информационих технологија - оглед, социјалним партнерима и свим другим институцијама и појединцима заинтересованим за ову област.

Будући да успешно спровођење матурског испита претпоставља припрему свих учесника и примену прописаних процедура, упутства из овог приручника су важна како би се осигурало да се испит на исти начин спроводи у свакој школи и да га сви ученици полажу под једнаким условима.

Овај документ ће у наредном периоду бити унапређиван и проширен у складу са захтевима и потребама четврогодишњих образовних профиле, школа и социјалних партнера.

¹ "Службени гласник РС" бр. 1/2007

² "Службени гласник РС" бр. 21/2009

³ "Службени гласник РС" бр. 107/2012

⁴ "Службени гласник РС – Просветни гласник" бр. 04/12;09/14;14/15;

НОВИ КОНЦЕПТ МАТУРСКОГ ИСПИТА

Матурски испит је један од елемената система обезбеђивања квалитета стручног образовања. Полагањем матурског испита у средњем стручном образовању, за разлику од општег средњег образовања, појединац стиче **квалификацију** неопходну за учешће на тржишту рада. Истовремено, положен матурски испит омогућава наставак даљег школовања у складу са законом.

Матурским испитом по новом концепту се проверава да ли је ученик, по успешно завршеном четвртогодишњем образовању, стекао програмом прописана знања, вештине ставове и способности, тј. главне стручне компетенције за занимање(а) за које се школовао у оквиру образовног профиле. Поред дипломе, сваки појединац полагањем оваквог испита стиче и тзв. додатак дипломи - **Уверење о положеним испитима у оквиру савладаног програма огледа за образовни профил**, чиме се на транспарентан начин послодавцима представљају стечене (проверене) компетенције.

Нови концепт матурског испита који се примењује у свим огледним одељењима истог образовног профиле заснован је на **принципима**:

- уједначавање квалитета матурског испита на републичком нивоу,
- увођење делимично екстерног оцењивања,
- унапређивање квалитета оцењивања.

Уједначавање квалитета матурског испита на републичком нивоу подразумева спровођење испита по једнаким захтевима и под једнаким условима у свим школама. Нови концепт матурског испита подразумева **тимски рад наставника** свих школа у којима се испит реализује, како у делу процеса креирања инструмената у коме сви наставници равноправно учествују, тако и у делу припремне наставе као наставници – ментори. На тај начин се доприноси уједначавању квалитета образовања на републичком нивоу за сваки образовни профил.

Увођење делимично екстерног оцењивања односи се на учешће представника послодавца - стручњака у одређеној области у оцењивању као екстерних чланова комисије и доприноси унапређивању методологије оцењивања у стручном образовању.

Унапређивање квалитета оцењивања постиже се, између остalog, интегративним и интердисциплинарним приступом у креирању тестова за проверу стручно-теоријских знања, као и употребом стандардизованих инструмената за процену стечених радних компетенција. Инструменти за оцењивање практичних радних задатака засновани су на операционализацији радних задатака проистеклих из реалних захтева посла и садрже јасно дефинисане мере процене, те представљају основ за стандардизацију испита у стручном образовању.

Резултати матурског испита могу се користити у процесу **самовредновања** квалитета рада школе, али и **вредновања** образовног процеса у датом образовном профилу, на националном нивоу. Они су истовремено и смерница за унапређивање образовног процеса на оба нивоа.

За сваки образовни профил припрема се **Приручник за полагање матурског испита** (у даљем тексту: Приручник), којим се детаљно уређује припрема, организација и реализација матурског испита.

Приручник садржи:

- Програм матурског испита
- Збирку теоријских задатака за матурски испит - Анекс 1
- Радне задатке са обрасцима за оцењивање на матурском практичном раду – Анекс 2

Приручнике припрема, у сарадњи са тимовима наставника, Завод за унапређивање образовања и васпитања – Центар за стручно образовање и образовање одраслих.

ПРОГРАМ МАТУРСКОГ ИСПИТА

1. ЦИЉ МАТУРСКОГ ИСПИТА

Матурским испитом проверава се да ли је ученик, по успешно завршеном образовању по програму огледа за образовни профил **електротехничар информационих технологија - оглед**, стекао програмом прописана знања, вештине и ставове и способности, односно стручне компетенције за занимање за које се школовао у оквиру образовног профила.

2. СТРУКТУРА МАТУРСКОГ ИСПИТА

Матурски испит за ученике који су се школовали по Наставном плану и програму за образовни профил електротехничар информационих технологија - оглед, састоји се из три дела:

- испит из материјег језика⁵ и књижевности,
- испит за проверу стручно–теоријских знања,
- матурски практични рад.

3. ОРГАНИЗАЦИЈА МАТУРСКОГ ИСПИТА

Матурски испит спроводи се у школи и просторима где се налазе радна места и услови за реализацију матурског практичног рада. Матурски испит се организује у три испитна рока која се реализују у јуну, августу и јануару.

За сваког ученика директор школе одређује ментора. Ментор је наставник стручних предмета који је обучавао ученика у току школовања. Он помаже ученику у припремама за полагање испита за проверу стручно – теоријских знања и матурског практичног рада.

У оквиру периода планираног наставним планом и програмом за припрему и полагање матурског испита, школа организује консултације и додатну припрему ученика за полагање испита, обезбеђујући услове у погледу простора, опреме и временског распореда.

Матурски испит за ученика може да траје четири дана. У истом дану ученик може да положе само један део матурског испита.

За сваки део матурског испита директор школе именује стручну испитну комисију, коју чине три члана, као и њихове замене. Сваки део матурског испита се оцењује и на основу тих оцена утврђује се општи успех на матурском испиту.

4. УСЛОВИ ЗА ПОЛАГАЊЕ МАТУРСКОГ ИСПИТА

Ученик положе матурски испит у складу са Законом.

Услови за полагање матурског испита дати су у табели бр. 1.

⁵Под материјум језиком подразумева се српски језик, односно језик националне мањине на коме се ученик школовао.

Табела бр. 1. Услови за полагање матурског испита

Предуслови за ученика:	
општи:	Успешно завршен четврти разред образовног профила електротехничар информационих технологија - оглед
посебни:	Прибор за писање и рачунање (обавезна хемијска оловка за тест знања) Одговарајући обрасци за унос и обраду података
Услови за школу:	
За припрему и спровођење матурског испита неопходно је да школа, самостално или у договору са социјалним партнерима, обезбеди потребне услове:	
<ul style="list-style-type: none"> • време (термине за извођење свих свих делова матурског испита, укључујући план реализације радних задатака) • простор за реализацију испита • за сваког ученика рачунар са: <ul style="list-style-type: none"> - прикључком за интернет - инсталираним развојним окружењем за развој десктопа и веб апликације - сервером базе података и потребним базама података - инсталираним локалним веб сервером, или приступом веб серверу унутар локалне мреже, или Интернета • записнике о полагању матурског испита за сваког ученика • одговарајући број примерака тестова • комбинације радних задатака за дати испитни рок • описе радних задатака и довољан број прилога који прате појединачне радне задатке, за сваког ученика и члана испитне комисије • обрасце за оцењивање радних задатака за сваког члана испитне комисије • чланове комисија обучене за оцењивање засновано на компетенцијама 	

Ученици који не задовољавају прописане услове не могу приступити полагању матурског испита.

5. ЕВИДЕНЦИЈА НА МАТУРСКОМ ИСПИТУ

Током матурског испита за сваког ученика појединачно, води се **записник** о полагању матурског испита. У оквиру записника прилажу се:

- писмени састав из материјег језика,
- оцењен тест са испита за проверу стручно - теоријских знања,
- стандардизовани обрасци за оцењивање сваког појединачног радног задатака за сваког члана комисије.

6. ДИПЛОМА И УВЕРЕЊЕ

Ученик који је положио матурски испит, стиче право на издавање *Дипломе о стеченом средњем образовању за образовни профил електротехничар информационих технологија - оглед*.

Уз диплому ученик добија и *Уверење о положеним испитима у оквиру савладаног програма огледа за образовни профил електротехничар информационих технологија - оглед*.

7. ИСПИТ ИЗ МАТЕРЊЕГ ЈЕЗИКА И КЊИЖЕВНОСТИ

Циљ испита је провера језичке писмености, познавања књижевности, као и опште културе.

7.1 СТРУКТУРА ИСПИТА

Испит из матерњег језика полаже се писмено.

На испиту ученик обраћајује једну од четири понуђене теме. Ове теме утврђује Испитни одбор школе, на предлог стручног већа наставника матерњег језика. Од четири теме које се нуде ученицима, две теме су из књижевности, а две теме су слободне.

7.2 ОЦЕЊИВАЊЕ

Оцену писаног рада утврђује испитна комисија на основу појединачних оцена сваког члана испитне комисије.

Испитну комисију за матерњи језик чине три наставника, од којих је најмање један наставник матерњег језика, и од којих се један именује за председника комисије. Сваки писмени састав прегледају сва три члана комисије и изводе јединствену оцену.

7.3 ОРГАНИЗАЦИЈА ИСПИТА

- Писмени испит из матерњег језика траје три сата.
- У току испита у свакој школској клупи седи само један ученик.
- За време израде писаног састава у ученици дежура наставник који није члан Стручног већа наставника матерњег језика.
- Дежурни наставник исписује називе одобраних тема на школској табли и од тог тренутка се рачуна време трајања испита.
- Дежурни наставник прикупља све ученичке радове и записнички их предаје председнику испитне комисије за матерњи језик.
- Након евидентираних и изведеных јединствених оцена за сваког од ученика председник испитне комисије сумира резултате и предаје потписане записнике и ученичке радове председнику Испитног одбора.

8. ИСПИТ ЗА ПРОВЕРУ СТРУЧНО-ТЕОРИЈСКИХ ЗНАЊА

Циљ овог дела матурског испита је провера остварености очекиваних исхода знања за образовни профил **електротехничар информационих технологија - оглед**, односно стручно-теоријских знања неопходних за обављање послова и задатака за чије се извршење ученик оспособљава током школовања.

8.1 СТРУКТУРА ИСПИТА

У наставном плану и програму огледа за образовни профил **електротехничар информационих технологија - оглед**, теоријска стручна знања неопходна за обављање послова и задатака електротехничара информационих технологија - оглед, стичу се у оквиру предмета:

- Програмирање
- Веб дизајн
- Веб програмирање
- Информациони системи и базе података

Провера остварености очекиваних исхода знања, односно стручно-теоријских знања врши се завршним тестирањем. Тест садржи **највише 50 задатака**. Комбинација задатака за матурски тест, узимајући у обзир и критеријум сазнајне сложености, формира се од задатака која се налазе у Збирци теоријских задатака (Анекс 2). Збирку су, уз координацију Центра, припремили чланови радне групе за развој програма матурског испита (координатор радне групе и представници школа у којима се школске 2015/2016. године реализује матурски испит и школских тимова).

Збирка задатака организована је у 4 области, које одговарају изучаваним предметима и стручним модулима, и садржи задатке којима се проверавају исходи знања тих предмета и модула.

Тест и кључ за оцењивање теста припрема Центар, на основу збирке теоријских задатака за матурски испит и доставља га школама.

Тест садржи познате задатке објављене у збирци (75 бодова) и делимично измене задатке из збирке (25 бодова).

Тест који ученици решавају садржи задатке којима се испитује достигнутост исхода учења прописаних наставним планом и програмом за образовни профил **електротехничар информационих технологија - оглед**. Тестови су конципирани тако да обухватају све нивое знања и све садржаје који су процењени као

темељни и од суштинског значаја за обављање послова и задатака у оквиру занимања као и за наставак школовања у матичној области.

Комисију за преглед тестова чине три наставника стручних предмета.

Заступљеност предмета у тесту дефинисала је радна група. Структура теста према областима дата је у табели бр. 2.

Табела бр. 2. Структура теста

област	процентуално учешће у тесту
1. Програмирање <ul style="list-style-type: none"> - Програмски језик Ц – 20% - Програмски језик Јава – 20% 	35 (+20%)
2. Веб дизајн	10
3. Веб програмирање	5
4. Информациони системи и базе података	30
Укупно	100

8.2 ОЦЕЊИВАЊЕ

Тестове прегледа трочлана комисија, коју чине наставници стручних предмета и модула, а према кључу достављеном из Центра. Сваки тест прегледају сва три члана комисије, о чему сведоче својим потписима на тесту.

Укупан бодова на тесту који ученик може да постигне је **100**, и једнак је збиру бодова које је ученик постигао тачним одговорима на постављене задатке. На тесту нема негативних бодова. Успех на тесту изражава се нумерички, при чему се број бодова преводи у успех, на основу скале за превођење бодова у успех, дате у табели бр. 3.

Табела бр. 3. Скала за превођење бодова у успех при оцењивању теста

УКУПАН БРОЈ БОДОВА	УСПЕХ
до 50 бодова	недовољан (1)
50,5 – 63	довољан (2)
63,5 – 75	добар (3)
75,5 – 87	врло добар (4)
87,5 - 100	одличан (5)

Утврђену нумеричку оцену комисија уноси на предвиђено место на обрасцу теста, и у Записник о полагању матурског испита.

8.3 ОРГАНИЗАЦИЈА ИСПИТА

- Тестирање у оквиру испита за проверу стручно–теоријских знања обавља се истовремено у свим школама у којима се реализује матурски испит за овај образовни профил. Термин тестирања утврђују школе и достављају га Центру најкасније месец дана пре реализације.
- По избору чланова комисије за преглед тестова, школе треба да изврше кратку обуку чланова комисије уз подршку стручних сарадника школе.
- Центар на основу утврђене структуре, формира тест и доставља га у електронској форми школама у којима се матурски испит реализује, 24 сата пре датума утврђеног за полагање теста, а кључ на дан реализације теста.

- Лице задужено за техничку припрему теста у школи обавља све припреме и умножава тест. Број копија већи је за три од броја ученика, како би сваки члан комисије добио по један примерак теста. Припремљени тестови се пакују у два коверта (један са тестовима за ученике, а други са 3 примерка за комисију која ће прегледати тест) који се затварају, печате и чувају у каси школе до почетка испита. За сигурност тестова, одговоран је директор школе.
- На дан испита, пола сата пре почетка, наставници дежурни током тестирања записнички преузимају коверат са тестовима за ученике и отпечаћују га у учионици, пред ученицима.
- Израда теста траје два сата. Током израде теста, сваки ученик седи сам у клупи. У учионици, где се врши тестирање, дежурају по два наставника, који према Правилнику о врсти образовања наставника у стручним школама не могу предавати предмете/модуле обухваћене тестом.
- За решавање теста ученик треба да користи хемијску оловку (коначни одговори и резултати морају бити исписани хемијском оловком).
- По завршетку теста, на огласној табли школе, објављује се кључ теста. Најкасније у року од 24 сата по завршетку реализације теста објављују се резултати тестирања на огласној табли школе.

9. МАТУРСКИ ПРАКТИЧНИ РАД

Циљ матурског практичног рада је провера главних радних (стручних) компетенција прописаних планом и програмом за образовни профил *Електротехничар информационих технологија - оглед*.

9.1 СТРУКТУРА ИСПИТА

На матурском практичном раду ученик извршава два радна задатка којима се проверавају прописане главне радне компетенције.

За проверу прописаних компетенција утврђује се **листа стандардизованих радних задатака**. Од стандардизованих радних задатака сачињава се одговарајући број комбинација радних задатака за матурски практични рад. Листе стандардизованих радних задатака, комбинације, критеријуми и обрасци за оцењивање саставни су део Приручника.

На основу листе комбинација из Приручника, формира се школска листа комбинација у сваком испитном року. Број комбинација у школској листи мора бити најмање за 10% већи од броја ученика у одељењу који полажу матурски практичан рад. Ученик извлачи комбинацију радних задатака на дан полагања матурског практичног рада.

Радне задатке за проверу компетенција припремају чланови радне групе за развој програма матурског испита у сарадњи са Центром и републичким координатором за огледни образовни профил **електротехничар информационих технологија - оглед**.

Листа радних задатака дата је у табели бр. 4.

Табела бр. 4. Радни задаци у оквиру матурског практичног рада

стручна компетенција	шифра	назив радног задатка
A ИЗРАДА ВЕБ СТРАНИЦЕ И ВЕБ АПЛИКАЦИЈЕ И ОБЈАВА НА ВЕБ СЕРВЕРУ	ЕИТ-А1	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ДОМАЋЕ ЖИВОТИЊЕ
	ЕИТ-А2	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ФОТО ГАЛЕРИЈА
	ЕИТ-А3	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА
	ЕИТ-А4	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СРБИЈА И ЊЕНИ СУСЕДИ
	ЕИТ-А5	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕЗЕРВАЦИЈА АУТОБУСКИХ КАРАТА
	ЕИТ-А6	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТЕЛЕФОНСКИ ИМЕНИК
	ЕИТ-А7	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕД ВОЖЊЕ
	ЕИТ-А8	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВЕБ ПРОДАВНИЦА
	ЕИТ-А9	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КОНВЕРТОР ТЕКСТА
	ЕИТ-А10	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КУРСНА ЛИСТА
	ЕИТ-А11	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – НУМЕРОЛОШКО СЛАГАЊЕ
	ЕИТ-А12	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КЊИГА УТИСАКА
	ЕИТ-А13	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЦД КАТАЛОГ
	ЕИТ-А14	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТВ ПРОГРАМ
	ЕИТ-А15	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВОЗНИ ПАРК
	ЕИТ-А16	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РАСПОРЕД ЧАСОВА
	ЕИТ-А17	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ ТЕСТ
	ЕИТ-А18	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА
	ЕИТ-А19	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ РЕЧНИК
	ЕИТ-А20	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СЛАЈДЕР ЗА СЛИКЕ (Image slider)
	ЕИТ-А21	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СОРТИРАЊЕ ТАБЕЛЕ
	ЕИТ-А22	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕФЕКТИ ТАБЕЛЕ
	ЕИТ-А23	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КАРТИЦЕ ЛОЈАЛНОСТИ
	ЕИТ-А24	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – УСПЕХ УЧЕНИКА
	ЕИТ-А25	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – БИБЛИОТЕКА
	ЕИТ-А26	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЗВУЦИ ИНСТРУМЕНТАТА
	ЕИТ-А27	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – БИОРИТАМ

стручна компетенција	шифра	назив радног задатка
Б ИЗРАДА ДЕСКТОП АПЛИКАЦИЈА, ПРОЈЕКТОВАЊЕ ЈЕДНОСТАВНЕ БАЗЕ ПОДАТКА И ПОВЕЗИВАЊЕ АПЛИКАЦИЈЕ СА БАЗОМ ПОДАТКА	ЕИТ-Б1	ИЗЛОЖБЕ ПАСА
	ЕИТ-Б2	ИЗЛОЖБЕ ПАСА
	ЕИТ-Б3	ДВД КОЛЕКЦИЈА
	ЕИТ-Б4	ДВД КОЛЕКЦИЈА
	ЕИТ-Б5	ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ
	ЕИТ-Б6	ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ
	ЕИТ-Б7	ТУРИСТИЧКА АГЕНЦИЈА
	ЕИТ-Б8	ТУРИСТИЧКА АГЕНЦИЈА
	ЕИТ-Б9	АНТИКВИТЕТИ И ЛОКАЦИЈЕ
	ЕИТ-Б10	АНТИКВИТЕТИ И ЛОКАЦИЈЕ
	ЕИТ-Б11	ПОЗОРИШНЕ ПРЕДСТАВЕ
	ЕИТ-Б12	ПОЗОРИШНЕ ПРЕДСТАВЕ
	ЕИТ-Б13	КОЛЕКЦИЈА КЊИГА
	ЕИТ-Б14	КОЛЕКЦИЈА КЊИГА
	ЕИТ-Б15	ШКОЛСКА БИБЛИОТЕКА
	ЕИТ-Б16	ШКОЛСКА БИБЛИОТЕКА
	ЕИТ-Б17	ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА
	ЕИТ-Б18	ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА
	ЕИТ-Б19	ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ
	ЕИТ-Б20	ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ
	ЕИТ-Б21	ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА
	ЕИТ-Б22	ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА
	ЕИТ-Б23	ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ
	ЕИТ-Б24	ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ
	ЕИТ-Б25	КОМУНИКАЦИЈА СТУДЕНТА И ПРОФЕСОРА
	ЕИТ-Б26	КОМУНИКАЦИЈА СТУДЕНТА И ПРОФЕСОРА
	ЕИТ-Б27	ДНЕВНИ БОРАВАК ДЕЦЕ
	ЕИТ-Б28	ДНЕВНИ БОРАВАК ДЕЦЕ
	ЕИТ-Б29	РЕЗЕРВАЦИЈА ХОТЕЛСКИХ СОБА
	ЕИТ-Б30	РЕЗЕРВАЦИЈА ХОТЕЛСКИХ СОБА
	ЕИТ-Б31	АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА
	ЕИТ-Б32	АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА
	ЕИТ-Б33	ГОЛФ КЛУБ
	ЕИТ-Б34	ГОЛФ КЛУБ
	ЕИТ-Б35	ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ
	ЕИТ-Б36	ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ
	ЕИТ-Б37	СЕОСКИ ТУРИЗАМ
	ЕИТ-Б38	СЕОСКИ ТУРИЗАМ

Од стандардизованих радних задатака **Центар сачињава одговарајући број комбинација** радних задатака за овај испит. Комбинације се формирају на основу прописаних компетенција и критеријума сложености – захтевности радног задатка. На матурском испиту ученик израђује два радна задатка у оквиру одговарајуће комбинације задатака, како је то наведено у следећој табели.

Листа комбинација дата је у табели бр. 5.

Табела бр. 5. Комбинације радних задатака за матурски практични рад

комбинација број	радни задаци	комбинација број	радни задаци	комбинација број	радни задаци
1	ЕИТ-А1 ЕИТ-Б1	2	ЕИТ-А1 ЕИТ-Б2	3	ЕИТ-А1 ЕИТ-Б3
4	ЕИТ-А1 ЕИТ-Б4	5	ЕИТ-А1 ЕИТ-Б5	6	ЕИТ-А1 ЕИТ-Б6
7	ЕИТ-А1 ЕИТ-Б7	8	ЕИТ-А1 ЕИТ-Б8	9	ЕИТ-А1 ЕИТ-Б9
10	ЕИТ-А1 ЕИТ-Б10	11	ЕИТ-А1 ЕИТ-Б11	12	ЕИТ-А1 ЕИТ-Б12
13	ЕИТ-А1 ЕИТ-Б13	14	ЕИТ-А1 ЕИТ-Б14	15	ЕИТ-А1 ЕИТ-Б15
16	ЕИТ-А1 ЕИТ-Б16	17	ЕИТ-А1 ЕИТ-Б17	18	ЕИТ-А1 ЕИТ-Б18
19	ЕИТ-А1 ЕИТ-Б19	20	ЕИТ-А1 ЕИТ-Б20	21	ЕИТ-А1 ЕИТ-Б21
22	ЕИТ-А1 ЕИТ-Б22	23	ЕИТ-А1 ЕИТ-Б23	24	ЕИТ-А1 ЕИТ-Б24
25	ЕИТ-А1 ЕИТ-Б25	26	ЕИТ-А1 ЕИТ-Б26	27	ЕИТ-А1 ЕИТ-Б27
28	ЕИТ-А1 ЕИТ-Б28	29	ЕИТ-А1 ЕИТ-Б29	30	ЕИТ-А1 ЕИТ-Б30
31	ЕИТ-А1 ЕИТ-Б31	32	ЕИТ-А1 ЕИТ-Б32	33	ЕИТ-А1 ЕИТ-Б33
34	ЕИТ-А1 ЕИТ-Б34	35	ЕИТ-А1 ЕИТ-Б35	36	ЕИТ-А1 ЕИТ-Б36
37	ЕИТ-А1 ЕИТ-Б37	38	ЕИТ-А1 ЕИТ-Б38	39	ЕИТ-А2 ЕИТ-Б1
40	ЕИТ-А2 ЕИТ-Б2	41	ЕИТ-А2 ЕИТ-Б3	42	ЕИТ-А2 ЕИТ-Б4
43	ЕИТ-А2 ЕИТ-Б5	44	ЕИТ-А2 ЕИТ-Б6	45	ЕИТ-А2 ЕИТ-Б7
46	ЕИТ-А2 ЕИТ-Б8	47	ЕИТ-А2 ЕИТ-Б9	48	ЕИТ-А2 ЕИТ-Б10
49	ЕИТ-А2 ЕИТ-Б11	50	ЕИТ-А2 ЕИТ-Б12	51	ЕИТ-А2 ЕИТ-Б13
52	ЕИТ-А2 ЕИТ-Б14	53	ЕИТ-А2 ЕИТ-Б15	54	ЕИТ-А2 ЕИТ-Б16
55	ЕИТ-А2 ЕИТ-Б17	56	ЕИТ-А2 ЕИТ-Б18	57	ЕИТ-А2 ЕИТ-Б19
58	ЕИТ-А2 ЕИТ-Б20	59	ЕИТ-А2 ЕИТ-Б21	60	ЕИТ-А2 ЕИТ-Б22
61	ЕИТ-А2 ЕИТ-Б23	62	ЕИТ-А2 ЕИТ-Б24	63	ЕИТ-А2 ЕИТ-Б25
64	ЕИТ-А2 ЕИТ-Б26	65	ЕИТ-А2 ЕИТ-Б27	66	ЕИТ-А2 ЕИТ-Б28
67	ЕИТ-А2 ЕИТ-Б29	68	ЕИТ-А2 ЕИТ-Б30	69	ЕИТ-А2 ЕИТ-Б31
70	ЕИТ-А2 ЕИТ-Б32	71	ЕИТ-А2 ЕИТ-Б33	72	ЕИТ-А2 ЕИТ-Б34
73	ЕИТ-А2 ЕИТ-Б35	74	ЕИТ-А2 ЕИТ-Б36	75	ЕИТ-А2 ЕИТ-Б37
76	ЕИТ-А2 ЕИТ-Б38	77	ЕИТ-А3 ЕИТ-Б1	78	ЕИТ-А3 ЕИТ-Б2
79	ЕИТ-А3 ЕИТ-Б3	80	ЕИТ-А3 ЕИТ-Б4	81	ЕИТ-А3 ЕИТ-Б5

У Табели бр. 5 наведена је само 81 комбинација од могућих свих 1026 комбинација радних задатака. Све потенцијалне комбинације су могуће на матурском раду, али због простора нису све и наведене у Табели бр. 5. Листу радних задатака, обрасце за оцењивање радних задатака, и листу комбинација, Центар доставља школама у оквиру овог Приручника.

9.2 ОЦЕЊИВАЊЕ

Оцену о стеченим стручним компетенцијама на матурском практичном раду даје **испитна комисија**.

Њу чине најмање три члана, које именује директор школе, према прописаној структури:

- два наставника стручних предмета за образовни профил електротехничар информационих технологија - оглед, од којих је један председник комисије
- представник послодавца – стручњак у области информационих технологија – кога предлаже Унија послодавца Србије у сарадњи са одговарајућим пословним удружењима, Привредном комором Србије и Центром⁶.

Време израде радних задатака дефинисано је у сваком задатку појединачно и контролишу га сви чланови комисије.

Након завршеног испита чланови комисије приступају прегледу радова ученика. Сваки члан комисије индивидуално оцењује рад ученика користећи одговарајући образац за оцењивање радног задатка.

Радни задатак може се оценити са највише **100 бодова**.

Сваки члан испитне комисије вреднујући појединачне индикаторе у свом обрасцу за оцењивање радног задатка утврђује укупан број бодова који је ученик остварио извршењем задатка. На основу појединачног бодовања свих чланова комисије утврђује се просечан број бодова за задатак.

Када кандидат оствари просечних 50 и више бодова, по сваком радном задатку, сматра се да је показао компетентност.

Ако је просечни број бодова на појединачном радном задатку који је кандидат остварио његовим извршењем, мањи од 50, сматра се да кандидат није показао компетентност. У овом случају оцена успеха на матурском практичном раду је недовољан (1).

Укупан број бодова које ученик оствари на матурском практичном раду једнак је збире постигнутих бодова на свим радним задацима, уз испуњен услов да је по задатку остварено најмање 50 бодова. Укупан број бодова преводи се у успех. Скала успешности је петостепена и приказана је у следећој табели.

Бодови се преводе у успех према следећој скали:

Табела бр. 6. Скала за превођење бодова у успех при оцењивању матурског практичног рада

Укупан број бодова -два радна задатка-	УСПЕХ
до 99 бодова	недовољан (1)
100-125	довољан (2)
126-151	добар (3)
152-177	врло добар (4)
178-200	одличан (5)

9.3 ОРГАНИЗАЦИЈА ИСПИТА

- Матурски практичан рад реализује се у школским рачунарским кабинетима, или просторима где се налазе радна места и услови за које се ученик образовао у току свог школовања.
- Стручно веће наставника стручних предмета школе бира комбинације радних задатака на основу листе комбинација из овог Приручника, и формира школску листу која ће се користити у том испитном року. Листа се формира у складу са могућностима реализације. Број комбинација мора бити за 10% већи од броја ученика који полажу матурски испит у једној школи.
- По формирању Испитног одбора директор утврђује чланове комисија за оцењивање матурског практичног рада и њихове заменике. Имена екстерних чланова комисије достављају се Центру.
- По избору чланова комисије за оцењивање, школа треба да изврши обуку чланова комисије уз подршку стручних сарадника школе.

⁶ Базу података о екстерним члановима испитних комисија води Центар

- Ученик извлачи радни задатак непосредно пре почетка самог испита. Ученик извлачи један од понуђених задатака без права замене.
- Лице задужено за техничку подршку реализацији матурског практичног рада припрема за сваког члана испитне комисије описе задатака, по један образац за оцењивање за сваки радни задатак у оквиру свих извучених комбинација задатака, збирне обрасце, као и одговарајуће прилоге за ученике и за чланове испитне комисије. У збирне обрасце за оцењивање се пре штампања уносе општи подаци о школи.
- Сваком ученику се обезбеђују једнаки услови за почетак обављања радног задатка.
- Најкасније у року од 24 сата по завршетку реализације практичног матурског рада објављују се резултати тог дела испита на огласној табли школе.

АНЕКС 1

ЗБИРКА ТЕОРИЈСКИХ ЗАДАТАКА ЗА МАТУРСКИ ИСПИТ

Драге ученице, драги ученици,

Пред вама је збирка задатака за завршно тестирање у оквиру матурског испита за огледни образовни профил **електротехничар информационих технологија - оглед**. Збирка је намењена вежбању и припремању за полагање испита за проверу стручно теоријских знања, и то из стручних предмета: **Програмирање, Веб дизајн, Веб програмирање и Информациони системи и базе података**.

У збирци се налазе задаци од којих ће се и формирати завршни тест знања.

Задаци у збирци распоређени су према областима, чији се исходи проверавају завршним тестом знања. У оквиру сваке области задаци су разврстани према облику задатка, а за сваки задатак је назначен број бодова који доноси.

Тест који ћете решавати на матурском испиту садржи задатке свих нивоа сложености којима се испитује оствареност исхода образовања за огледни образовни профил **електротехничар информационих технологија - оглед**. На тесту нема негативних бодова. Задаци носе различити број бодова у зависности од тога колико информација се тражи и колико треба да будете мисаоно ангажовани када одговарате. Важно је да пажљиво одговарате на задатке, јер сваки тачан одговор носи од 0,5 до 1 бода, а свака грешка аутоматски 0 бодова за задатак у целости. Збирка задатака не садржи решења.

Збирку задатака су израдили тимови наставника из свих школа у Републици Србији у којима се реализује матурски испит школске 2015/2016. године за огледни образовни профил **електротехничар информационих технологија - оглед**, у сарадњи са стручњацима Завода за унапређивање образовања и васпитања.

Желимо вам срећан и успешан рад!

ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК С

У следећим задацима заокружите број испред траженог одговора

1. Декларисане су две реалне променљиве које представљају реални и имагинарни део комплексног броја `float Re, Im;`

Кроз извршавање програма ове променљиве могу добити било коју (позитивну или негативну) вредност. Заокружити број испред наредбе приказа која ће приказати комплексни број у облику: $Z = 1.50 +3.20 j$ тј. $Z = -1.50 -3.20 j$ (у зависности од знака реалног и имагинарног дела броја)

2

1. `printf("Z = %.2f %+.2f j", re, im);`
2. `printf("Z = %.2f +%.2f j", re, im);`
3. `printf("Z = %+.2f %+ .2f j", re, im);`
4. `printf("Z = %-.2f %-.2f j", re, im);`

2. Дата је наредба декларације, а затим и наредба форматираног излаза:

```
float x = 5.56;
printf(" x = %f\nx = %g\n", x, x);
```

Након извршења ових наредби на екрану ће се приказати вредности променљивих у задатом формату. Заокружити број испред тачног одговора:

2

1. $x = 5.560000e+000 \quad x= 0$
2. $x = 5.560000 \quad x = 5.560000e+000$
3. $x = 5.560000 \quad x = 5.56$
4. $x = 5.56 \quad x = 5.560000e+000$

3. Заокружити број испред очекиваног одговора.

У програмски језику Ц подаци су предмет обраде. Сваки податак има одређене особине које чине тип податка. Тип податка је одређен и дефинисан:

1

1. скупом могућих вредности које може да узме податак
2. скупом операција које могу да се изврше над подацима
3. скупом могућих вредности које може да узме податак и скупом операција које могу да се изврше над подацима

4. Дата је декларација набројивог типа податка **boja**:

```
enum boja {crna, plava, zelena, crvena=4, bela=15};
```

Имајући у виду дефиницију набројивог типа податка, одредити вредности које имају константе crna, plava, zelena, crvena и bela. Заокружити број испред очекиваног одговора:

1

1. $crna = 0, plava = 1, zelena=2, crvena = 4, bela = 15$
2. $crna =1, plava = 2, zelena=3, crvena = 4, bela = 15$
3. $crna =255, plava = 254, zelena=253, crvena = 4, bela = 15$

5. Декларисана је целобројна променљива и додељена јој је вредност логичког израза:

```
int x;
x = 1==10>5;
```

Имајући у виду приоритет оператора, одредити вредност променљиве x после извршења ове наредбе. Заокружити број испред траженог одговора:

2

1. променљива добија вредност логичке неистине, тј. $x = 0$
2. променљива добија вредност логичке истине, тј. $x = 1$
3. вредност логичког израза се не може доделити целобројној променљивој
4. променљива добија вредност логичке истине, тј. било који број различит од 0

6. Декларисане су две целобројне променљиве. Променљива **a** добија вредност уносом са тастатуре. Затим се вредност логичког израза додељује променљивој **x**:

```
int x;
scanf("%d", &a);
x = 10!=5 || a<2;
```

Имајући у виду приоритет оператора, одредити вредност променљиве x после извршења ове наредбе. Заокружити број испред траженог одговора:

2

1. уколико се заградама не одредити редослед извршавања операција у овом изразу, долази до грешке, тј. „пуцања“ програма
2. без обзира на вредност која се унесе у променљиву **a**, вредност израза је увек „тачно“, тј. **x = 1**
3. без обзира на вредност која се унесе у променљиву **a**, вредност израза је увек „нетачно“, тј. **x = 0**
4. вредност израза зависи од променљиве **a** и не може се једнозначно одредити уколико није позната вредност уписана у променљиву **a**

7. У следећем задатку заокружити број испред траженог одговора. Проценити шта ради следећи део Ц кода са подацима a, b, c и d.

```
a = a + (b+c+d);
b = a - (b+c+d);
c = a - (b+c+d);
d = a - (b+c+d);
a = a - (b+c+d);
printf("%d%d %d %d ", a, b, c, d);
```

2

1. Сортира податке у растућем редоследу.
2. Сортира податке у опадајућем редоследу
3. Врши замену података цикличним помаком у десно
4. Врши замену података цикличним помаком у лево

8. У следећем задатку заокружити број испред траженог одговора.

Дата је наредба у Ц језику. Наредба проверава да ли је број паран или непаран. Проценити која наредба даје тачно решење.

1. (broj % 2 == 1) ? printf("PARAN!!") : printf("NEPARAN!!");
2. (broj % 2) ? printf("PARAN!!") : printf("NEPARAN!!");
3. (broj % 2 == 0) ? printf("PARAN!!") : printf("NEPARAN!!");
4. (broj & 1) ? printf("PARAN!!") : printf("NEPARAN!!");
5. (broj & 0x1 == 0) ? printf("PARAN!!") : printf("NEPARAN!!");
6. (broj & 1 == 1) ? printf("PARAN!!") : printf("NEPARAN!!");

2

9. Дат је део кода на програмском језику Ц. Анализирати код и одредити које вредности целобројних података а и б ће се исписати након извршења следећих наредби за излаз. Заокружити број испред траженог одговора:

```
int a = 10, b = 2;
printf ("a = %d b = %d\n", a, b);
printf ("a-- -b = %d\n", a-- -b);
printf ("a = %d b = %d\n", a, b);
printf ("++a+b++ = %d\n", ++a + b++);
printf ("a = %d b = %d\n", a, b);
```

- | | | |
|---|--|---|
| 1. a = 10 b = 2
a-- -b = 9
a = 9 b = 3
++a + b++ = 11
a = 10 b = 3 | 2. a = 10 b = 2
a-- -b = 7
a = 9 b = 2
++a + b++ = 12
a = 10 b = 3 | 2 |
| 3. a = 11 b = 2
a-- -b = 8
a = 10 b = 3
++a + b++ = 13
a = 10 b = 4 | 4. a = 10 b = 2
a-- -b = 8
a = 9 b = 2
++a + b++ = 12
a = 10 b = 3 | |

10. Програмски језик Ц има у својим библиотекама функције које могу да се користе позивањем за одређене параметре. Функција **rand()** из библиотеке **stdlib.h** се често користи као генератор случајних бројева. Дат је код:

```
int slucajniBroj, broj;
slucajniBroj = rand() % broj;
```

Заокружити број испред исказа који дефинише опсег генерираног случајног броја:

1

1. Функција генерише случајан број са интервала 1 до (број).
2. Функција генерише случајан број са интервала 0 до (број - 1).
3. Функција генерише случајан број са интервала 0 до (број).
4. Функција генерише случајан број са интервала 1 до (број - 1).

11. Дата је декларација променљивих **int a, b, c;**

Израчунати вредности коју ће променљиве имати по извршењу следећих наредби:

```
a = c = 6;
b = 7;
a = b++ - 2;
c = ++b - 2;
```

Заокружити број испред очекиваног одговора:

1

1. a = 5 b = 9 c = 7
2. a = 4 b = 6 c = 7
3. a = 4 b = 5 c = 8
4. a = 6 b = 6 c = 6

12. Заокружити број испред кључне речи којом се при кодирању у програмском језику Ц, у наредби вишеструког гранања обележавају вредности за које се улази у поједине гране:

1. switch
2. break
3. return
4. case

1

13. Дат је код на програмском језику Ц:

```
switch(c) {
    case 'P': case 'p': printf("Pravougaonik ");
    case 'T': case 't': printf("Trougao ");
    case 'K': case 'k': printf("Krug ");
    default: printf("Nije geometrijska figura ");
}
```

Имајући у виду да је променљивој char с додељена је вредност '**T**', заокружити број испред понуђеног одговора који представља садржај приказан на екрану по извршењу датог програмског сегмента:

1. Trougao
2. Trougao Krug
3. Trougao Krug Nije geometrijska figura
4. Pravougaonik Trougao Krug Nije geometrijska figura

2

14. У програмском језику Ц декларисане су две целобројне променљиве int k, p; Следећу switch структуру реализовати помоћу if структуре.

```
switch(k) {
    case 1: case 3: case 5: p-=1; break;
    case 4: p=p-2; break;
}
```

Заокружити број испред траженог одговора:

1. **if**(k==1 && k==3 && k==4 && k==5) p--;
2. **if**(k==1 || k==3 || k==5) p--; **else if**(k==4) p-=2;
3. **if**(k=1 || k=3 || k=5) p--; **else if**(k=4) p-=2;
4. **if**(k==1 || k==3 || k==4 || k==5) p--;

2

15. У програмском језику Ц декларисане су две целобројне променљиве int k, p; Следећу switch структуру реализовати помоћу if структуре.

```
switch(k) {
    case 1: case 3: case 5: p-=1; break;
    case 4: p=p-1; break;
}
```

Заокружити број испред траженог одговора:

1. **if**(k==1 && k==3 && k==4 && k==5) p--;
2. **if**(k==1 || k==3 || k==4 || k==5) p--;
3. **if**(k=1 || k=3 || k=4 || k=5) p--;
4. **if**(k=1 && k=3 && k=4 && k=5) p--;

2

16. У програмском језику Ц декларисане су две целобројне променљиве int k, p; Следећу switch структуру реализовати помоћу if структуре.

```
switch(k) {
    case 1: case 3: case 5: p-=1;
    case 4: p=p-1;
}
```

Заокружити број испред траженог одговора:

1. **if**(k==1 && k==3 && k==4 && k==5) p--;
2. **if**(k==1 || k==3 || k==5) p-=2; **else if**(k==4) p-=1;
3. **if**(k=1 || k=3 || k=5) p--; **else if**(k=4) p--;
4. **if**(k==1 || k==3 || k==4 || k==5) p--;

2

17. У следећем задатку заокружити број испред траженог одговора.
Дат је део Ц кода који одређује вредност максимума између два целобројна податка и приказује га на екрану.
1. max=b;
`if(max<a) max=a;
printf("\nMaksimum = %d", &max);`
 2. b=max;
`if(max<a) max=a;
printf("\nMaksimum = %d", &max);`2
 3. max=b;
`if(max<a) max=a;
printf("\nMaksimum = %d", max);`
 4. max=a;
`if(b<max) max=b;
printf("\nMaksimum = %d", max);`
-
18. Код дат у тексту задатка треба реализовати помоћу једне if наредбе. Заокружити број испред понуђеног тачног одговора:
- ```
if(x>1)
{
 if(x<6)
 y=4;
}
```
1. `if(x>1 && x<6) y=4;`
  2. `if(x>1 || x<6) y=4;`
  3. `if(x<1 || x>6) y=4;`
  4. `if(!(x<=1 || x>=6)) y=4;`
- 
19. Дата је if-else наредба:
- ```
if(a==3 || a==5) p++;
else p--;
```
- Заокружити број испред кода који представља дату структуру приказану помоћу једне switch наредбе:
1. `switch(a) {
 case 3, 5: p++; break;
 default: p--;
}`
 2. `switch(a) {
 case 3: case 5: p++; break;
 p--;
}`1
 3. `switch(a) {
 case 3: case 5: p++; break;
 default: p--;
}`
 4. `switch(a) {
 case 3: case 5: p++;
 default: p--;
}`

20. Дат је део кода на програмском језику Ц који контролише унос карактер променљиве **izbor**. Анализирати шта се дешава ако корисник у моменту када програм затражи унос избора, укуца слово **d**. Заокружити број испред одговора који описује реакцију кода:

```
char izbor;
printf("Za izbor unesite D ili N: ");
do{
    izbor = getch();
}while(izbor!= 'D' && izbor!= 'N');
putch(izbor);
```

2

1. На екрану се приказује унето слово **d**, излази се из петље и наставља са извршењем програма
2. На екрану се приказује унето слово **d**, али се не излази из петље већ се чека унос слова **D** или **N**
3. На екрану се не приказује ништа и програм се понаша као да „не реагује“ на унос слова **d**
4. На екрану се не приказује унето слово, већ само порука којом се тражи поновни унос

21. Дат је део кода на програмском језику Ц којим је потребно реализовати унос текста знак по знак са тастатуре до задавања знака за крај стандардног улаза (CTRL+Z). Код треба да преброји унете знаке не узимајући у обзир знак за крај. Заокружити број испред наредбе коју треба убацити на ЈЕДНУ од означених позиција у телу петље да би код био функционалан:

```
char slovo; int brojac=0;
while(1) {
    a) _____
    b) slovo = getchar();
    c) _____
    d) brojac++;
}
```

2

1. Наредбу **if(slovo==NULL) break;** уметнути у линију c)
2. Наредбу **if(slovo==EOF) break;** уметнути у линију c)
3. Наредбу **if(slovo==EOF) continue;** уметнути у линију a)
4. Наредбу **if(slovo==NULL) continue;** уметнути у линију c)
5. Наредбу **if(slovo==EOF) break;** уметнути у линију a)

22. Дат је део кода на програмском језику Ц којим је потребно реализовати унос и пробројавање унетих цифара. Унос понављати све док не heiје први знак који није цифра. Заокружити број испред наредбе коју треба убацити на означене позиције у код да би он био функционалан:

```
char cifra;
int brojac=0;
while(1)
{
    cifra = getch();
    if(_____) { putch('\'n'); _____ }
    putch(cifra);
    brojac++;
}
```

2

1. На прву линију: **cifra< 0 || cifra> 9**, а на другу: **break;**
2. На прву линију: **cifra<'0' && cifra>'9'**, а на другу: **break;**
3. На прву линију: **cifra>='0' && cifra<='9'**, а на другу: **continue;**
4. На прву линију: **cifra<'0' || cifra>'9'**, а на другу: **break;**
5. На прву линију: **cifra<'0' || cifra>'9'**, а на другу: **continue;**

23. Дат је део кода на програмском језику Ц:

```
for(j=0; j<n; j++)
    if(a[j]>0) s+=a[j];
    else break;
```

Свака for петља може се написати коришћењем while и do-while наредбе. Заокружити број испред понуђеног кода који је еквивалентан коду датом у тексту задатка:

1.

```
j=0;
while(j<n && a[j]>0) s+=a[j++];
```
2.

```
j=0;
while(j<n && a[j++]>0) s+=a[j];
```
3.

```
j=0;
while(j<n || a[j]>0) s+=a[j++];
```
4.

```
j=0;
while(j<n && a[j]<=0) s+=a[j++];
```

2

24. Дата је декларација променљивих `unsigned a, b, x` и део кода у програмском језику Ц. Одредити шта се налази као резултат у променљивој `x` након извршења дате наредбе. Заокружити број испред траженог одговора:

```
unsigned a, b, x;
x=0;
while(a>=b) {
    a-=b;
    x++;
}
```

2

1. Производ `a` и `b`
2. Збир `a` и `b`
3. Остатак приликом делења
4. Количник при дељењу `a` са `b`

25. Дата је декларација променљивих `unsigned a, b` и део кода у програмском језику Ц. Одредити шта се налази као резултат у променљивој `x` и у након извршења датог кода. Заокружити број испред траженог одговора:

```
unsigned a, b, x, y, temp;
x=a*b;
while(b) temp=a%b, a=b, b=temp;
y=b;
x/=y;
```

2

1. `x` је производ `a` и `b`, а `y` је количник `a` са `b`
2. `x` је најмањи заједнички садржалац за `a` и `b`, а `y` највећи заједнички делилац за `a` и `b`
3. `x` је највећи заједнички делилац за `a` и `b`, а `y` најмањи заједнички садржалац за `a` и `b`
4. без обзира на вредности променљивих, долази до грешке у последњој наредби кода
5. долази до грешке јер петља понавља само прву наредбу услед изостанка витичастих заграда на телу петље

26. Дат је Ц код који након извршавања исцртава слику помоћу звездица.

```
int i,j,n=7;
for(i=1;i<=n/2;i++) {
    for(j=1;j<=n/2-i+1;j++) printf("  ");
    printf("*");
    for(j=1;j<=2*(i-1);j++) printf("  ");
    printf("\b*\b\n");
}
for(i=1;i<=n;i++) printf("* ");
```

Анализирати дати код и одредити која слика ће бити исцртана након његовог извршавања.
Заокружити број испред тачног одговора.

1. * * * * * *

Primer za n = 7

Primer za $n = 7$

3.  Primer za $n = 7$

4.  Primer za $n = 7$

5. * * * * * * *
 * *
 * *

27. Дата је декларација променљивих pod, br и део кода у програмском језику Ц. Закључити шта представља вредност коју променљива br добије извршењем кода:

```
unsigned pod, br;  
pod=128;  
br=0;  
while(pod!=0) {  
    if(pod & 0x1) br++;  
    pod>>=0x1;  
}
```

2

Заокружити број испред траженог одговора:

1. Број јединица у бинарном запису броја pod
 2. Број нула у бинарном запису броја pod
 3. Број цифара у бинарном запису броја pod
 4. Број цифара у хексадецималном запису броја pod

28. Програмски језик Ц има наредбе које омогућавају промену тока управљања у неку другу тачку програма. Једна од наредби за промену тока извршења програма, омогућава:

- превремено завршавање циклуса (*for*, *while* или *do-while*) скакањем на прву наредбу непосредно иза циклуса,
- прескацање следећих наредби унутар селекције *switch*
- очување структурности програма изван циклуса.

Заокружити број испред наредбе помоћу које се постижу горе наведени искази:

1. *continue*
2. *gotoxy*
3. *break*
4. *sizeof*

1

29. Једна од наредби за промену тока извршења програма у програмском језику Ц, омогућава:

- у случају када се налази унутар наредби ***while*** и ***do-while***, прелазак на поновно испитивање услова циклуса
- у случају наредбе ***for*** прелазак на извршавање израза 2 (тј. услова)
- у случају угњеждених циклуса, прескачу се само преостале наредбе најдубљег циклуса.
- наредба не нарушава структурираност програма ван циклуса
- ако се налази унутар наредбе ***switch*** која се налази унутар неког циклуса, скок се врши на крај тог циклуса уз наравно прескацање наредби које су биле унутар селекције ***switch***.

1

Заокружити број испред наредбе за коју важе горе наведени искази:

1. *continue*
2. *gotoxy*
3. *break*
4. *sizeof*

30. Заокружити број испред траженог одговора:

Дат је део кода на програмском језику Ц, који контролише унос целобројне променљиве *n*. Одредити вредности које променљива *n* може добити.

```
do {
    printf("Unesite N:\nN = ");
    scanf("%d", &n);
    if(n & 1) printf("Greska.\n");
}while(n & 1);
```

2

1. Омогућава унос непарног природног броја
2. Омогућава унос само позитивног природног броја
3. Омогућава унос само негативног природног броја
4. Омогућава унос парног природног броја
5. Омогућава унос само непарног позитивног природног броја

31. У програмском језику Ц, декларисан је и иницијализован низ целих бројева:

```
int A[7] = {100, 150, 200, 252, 300, 350, 400};
```

Низ се претражује методом **бинарне претраге**. Тражена вредност је **300**. Одредити број приступа низу (број покушаја) потребних да се пронађе тражена вредност и заокружити број испред траженог одговора:

1. два
2. три
3. пет
4. седам

2

32. Дата је декларација променљивих и део програмског кода:

```
int i, j, n, x[100], temp;
i=0;
j=n-1;
while(i<j) temp=x[i], x[i]=x[j], x[j]=temp, i++, j--;
```

Након извршења while циклуса низ x је преуређен. Проценити шта ради овај циклус. Заокружити број испред очекиваног одговора:

2

1. Сортира низ x од n елемената у растућем редоследу
2. Сортира низ x од n елемената у опадајућем редоследу
3. Инвертује (обрће) елементе низа x од n елемената
4. Ротира низ x од n елемената за једно место у лево
5. Ротира елементе низа x од n елемената за једно место у десно

33. Дата је декларација променљивих и део програмског кода:

```
int j, m, x[100], prom;
prom=0;
for(j=m-1; j>=0; j--) prom+=a[j];
```

Описати шта ради for циклус са елементима низа x. Заокружити број испред очекиваног одговора:

1

1. Сортира низ x од n елемената у растућем редоследу
2. Сабира елементе низа x од n елемената
3. Инвертује (обрће) елементе низа x од n елемената
4. Одређује број позитивних елемената низ x од n елемената
5. Ротира елементе низа x од n елемената за једно место у десно

34. Дата је декларација променљивих и део програмског кода:

```
int x[40] ={5, 8, 1, 2, 2, 6, 6, 6, 5};
int br=0, i, n, j;
for(i=0; i<n; i++) {
    j=0;
    while(j<i && x[j]!=x[i]) j++;
    if(i==j) br++;
}
```

2

Израчунати вредност коју ће променљива br имати по извршењу for циклуса. Заокружити број испред очекиваног одговора:

1. 7
2. 10
3. 5
4. 6

35. Дата је декларација променљивих и део програмског кода:

```
int i, temp, n = 11;
int x[30]={ -3, -1, -2, -2, 1, 4, 3, 1, 5, -8, 5 };
temp=x[0];
i=0;
while(i<n-1) x[i++]=x[i+1];
x[n-1]=temp;
```

Просудити на основу наредби које ће бити извршене у **while** циклусу како ће изгледати трансформисан низ **x** од **n** елемената. Заокружити број испред очекиваног одговора:

1. **x[] ={ 5, -3, -1, -2, -2, 1, 4, 3, 1, 5, -8 }**
2. **x[] ={ -1, -2, -2, 1, 4, 3, 1, 5, -8, 5, -3 }**
3. **x[] ={ -2, 0, -1, -1, 2, 5, 4, 2, 6, -7, 6 }**
4. **x[] ={ -1, -2, -2, 1, 4, 3, 1, 5, -8, 5 }**

2

36. Сортирање представља процес преуређивања скупа података по неком утврђеном поретку. Дат је део кода на програмском језику Ц који представља један од алгоритама за сортирање. Анализирати дати код и од понуђених одговора изабрати ком алгоритму сортирања припада. Заокружити број испред понуђеног тачног одговора:

```
for(i=1;i<n;i++) {
    temp=a[i];
    j=i-1;
    while(j>=0 && a[j]>temp) a[j+1]=a[j--];
    a[j+1]=temp;
}
```

2

1. selection sort
2. shell sort
3. bubble sort
4. compare selection sort
5. insertion sort

37. Ако се наредбом **p++** вредност показивача **p** увећа за 2, заокружити број испред тачне декларације показивача **p**:

1. **char *p;**
2. **short *p;**
3. **long *p;**
4. **float *p;**

1

38. Наредбом програмског језика Ц декларисан је показивач **p** као показивач на цео број коме је у наставку кода додељен је простор за смештање вектора целих бројева дужине **n**.

Заокружити број испред исправно написане наредбе којом се простор на који показује показивач **p** смањује за 2 елемента.

1. **p=malloc(p, sizeof(int)*(n-2));**
2. **malloc(p,sizeof(int)*(n-2));**
3. **p=realloc(p, n-2);**
4. **p=realloc(p, sizeof(int)*(n-2));**
5. **realloc(p, n-2);**
6. **realloc(p, sizeof(int)*(n-2));**

2

39. Дат је део кода написан у програмском језику Ц. Са леве стране је дат редни број линије кода, а са десне садржај линије кода. Заокружити број испред одговора у коме су наведени редни бројеви линија кода које ће иззврати грешку било приликом компајлирања или при извршењу кода:

Редни број линије кода	Линија кода	
a)	int *poc1, a=10;	
b)	double *poc2;	
c)	poc1=&8;	
d)	poc1=&a;	
e)	poc2=poc1;	

Заокружи број испред одговарајућег одговора:

1. 1, 2, 3
2. 3, 5
3. 1, 3, 5
4. 2, 5

2

40. У програмском језику Ц дана је наредба која декларише и иницијализује променљиве **a**, **b** и **c**.

```
struct elem1{
    int x, y;
};

struct elem{
    struct elem1 poz;
    int oblik;
};

struct elem a = {{1,5},3}, *b, c;
b=malloc(sizeof(struct elem));
```

2

Заокружити број испред одговора у ком су наредбе доделе између декларисаних променљивих **НЕТАЧНО** написане:

- | | |
|---|--|
| 1. c.poz.x = a.poz.x;
c.poz.y = a.poz.y;
c.oblik = a.oblik; | 2. b->poz.x = a.poz.x;
b->poz.y = a.poz.y;
b->oblik = a.oblik; |
| 3. c = a; | 4. b=a; |
| 5. b=&a; | 6. *b=a; |

41. Дат је део кода у програмском језику Ц. Проценити који од наведених исказа тачно одређује шта ће се десити након његовог извршавања.

```
int n=10, *poc, *x = (int*)malloc(n*sizeof(int));
for(poc=x; poc<x+n; poc++) *poc=0;
```

Заокружити број испред очекиваног одговора:

1. Извршавањем кода статички се резервише (алоцира) меморија за смештање низа и врши се иницијализација проласком кроз низ помоћу показивача.
2. Извршавањем кода дефинише се низ у динамичкој зони меморије и иницијализује проласком кроз низ помоћу индекса.
3. Извршавањем кода дефинише се низ у динамичкој зони меморије и иницијализује проласком кроз низ помоћу показивача.
4. Извршавањем кода статички се алоцира меморија за низ и врши се иницијализација проласком кроз низ помоћу индекса.

1

42. У програмском језику Ц дефинисана је показивачка променљива. Заокружи број испред реченице која даје тачну дефиницију показивачке променљиве:
1. Показивач је променљива која садржи вредност друге променљиве, константе или објекта.
 2. Показивач је променљива која садржи адресу друге променљиве, објекта или функције.
 3. Показивач је променљива која представља алијас друге променљиве, објекта или функције.
 4. Показивач је сложени тип података који у себи садржи адресе променљивих на које показује.
43. Дат је део кода у програмском језику Ц који представља дефиницију функције са прототипом `int *trazi(int x[], int n)`. Процени шта враћа дефинисана функција.
- ```
int *trazi(int x[], int n)
{
 int *poc=x;
 while(poc<x+n && (*poc<10 || *poc>99)) poc++;
 if(poc==x+n) return NULL;
 else return poc;
}
```
- Заокружити број испред исказа који тачно описује повратну вредност функције:
1. Враћа вредност првог двоцифреног броја низа x од n елемената, а ако двоцифреног броја нема враћа `NULL`.
  2. Враћа вредност последњег позитивног двоцифреног броја низа x од n елемената, а ако нема позитивних двоцифрених бројева, враћа `NULL`.
  3. Враћа адресу првог позитивног двоцифреног броја низа x од n елемената, а ако нема позитивних двоцифрених бројева, враћа показивач на почетак низа.
  4. Враћа адресу првог позитивног двоцифреног броја низа x од n елемената, а ако низ нема позитивних двоцифрених бројева, враћа `NULL`.
  5. Враћа адресу последњег позитивног двоцифреног броја у низу x од n елемената или `NULL` ако низ нема позитивних двоцифрених бројева.
44. Дат је следећи код написан у програмском језику Ц:
- ```
void main()
{
    int niz[10], *a, *b, *c, i;
    a = c = niz;
    b = a + 9;
    for(i=0; i < sizeof(niz)/sizeof(int); i++) a[i]=i;
    while(a < b) {
        if(*b & 1){ int c; c=*a; *a=*b; *b=c; }
        a++; b--;
    }
    for(i=0; i<10; i++) printf("%d ", *(c+i));
}
```
- Анализирати дати код и заокружити број испред одговора који описује шта ће се исписати на екрану након његовог извршавања:
1. 0 1 2 3 4 5 6 7 8 9
 2. 9 1 7 3 5 4 6 2 8 0
 3. 9 8 7 6 5 4 3 2 1 0
 4. 0 8 2 6 4 5 3 7 1 9

45. У програму написаном на програмском језику Ц дата је декларација дводимензионалног поља и његова иницијализација:

```
unsigned a[3][4]={{2, 4},{4, 6, 7, 3},{2, 1, 3}};
```

Одредити које вредности ће се доделити елементима декларисане матрице а.

Заокружити број испред очекиваног одговора:

1.
$$\begin{bmatrix} 0 & 0 & 2 & 4 \\ 4 & 6 & 7 & 3 \\ 0 & 2 & 1 & 3 \end{bmatrix}_{3 \times 4}$$

2.
$$\begin{bmatrix} 2 & 4 & 4 & 6 \\ 7 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{3 \times 4}$$

1

3.
$$\begin{bmatrix} 2 & 4 & 0 & 0 \\ 4 & 6 & 7 & 3 \\ 2 & 1 & 3 & 0 \end{bmatrix}_{3 \times 4}$$

4.
$$\begin{bmatrix} 2 & 0 & 0 & 4 \\ 4 & 6 & 7 & 3 \\ 2 & 1 & 0 & 3 \end{bmatrix}_{3 \times 4}$$

1

46. У програму написаном у програмском језику Ц дата је декларација дводимензионалног поља:

```
unsigned i, j, a[MAXV][MAXK];
```

Одредити везу између елемента матрице а са индексима i и j и вредности показивача који показује на тај елемент. Заокружити број испред везе која је исправно написана:

1. $a[i][j] \Leftrightarrow * (a + MAXK*j + i)$
2. $a[i][j] \Leftrightarrow * (a + MAXV*i + j)$
3. $a[i][j] \Leftrightarrow * ((\text{unsigned } *)a + MAXK*i + j)$
4. $a[i][j] \Leftrightarrow * ((\text{unsigned } *)a + MAXV*j + i)$

47. У програмском језику Ц је дата декларација променљивих, а касније у коду извршен позив функције на следећи начин:

```
int k, i;
char lista[10][50], ime[50];
if( Formiraj(lista[i], ime, k) == NULL) { ... }
```

На основу позива, проценити каквог је облика прототип функције и заокружити број испред тачно написаног прототипа:

2

1. `void *Formiraj(char s1, char s2, int x);`
2. `char Formiraj(char *s1, char *s2, int x);`
3. `int *Formiraj(char s1[], char s2[], int x);`
4. `int Formiraj(char s1[], char s2[], int x);`
5. `char *Formiraj(char s1, char s2, int x);`

48. У програмском језику Ц је дата декларација променљивих, а касније у коду извршен позив функције на следећи начин:

```
int x, y, i, j;
float **mat, *vek, z;
mat[i] = Formiraj(x, 0.5);
```

На основу позива, проценити каквог је облика прототип функције и заокружити број испред тачно написаног прототипа:

2

1. float Formiraj(int n, float m);
2. void *Formiraj(int n, int m);
3. float *Formiraj(float n, int m);
4. float *Formiraj(int n, float m);
5. float **Formiraj(int n, float m);

49. Дат је програм написан на програмском језику Ц који садржи функцију `void zameni(int a, int b)` и функцију `main()` у којој су декларисане променљиве које се користе као стварни параметри при позиву дате функције. Анализирати код и проценити које вредности променљивих `a` и `b` ће бити исписане након извршења програма.

1

```
void zameni(int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
int main()
{
    int a = 3, b = 4;
    zameni(a, b);
    printf("\n a=%d i b=%d", a, b);
}
```

Заокружити број испред тачног одговора:

1. a=3 i b=3
2. a=4 i b=4
3. a=3 i b=4
4. a=4 i b=3

50. На левој страни је дата дефиниција функције `void zameni(int *a, int *b)`, а на десној страни део кода написан на програмском језику Ц, који позива дефинисану функцију.

1

```
void zameni(int *a, int *b)      int a = 3, b = 4;
{
    int temp;                  zameni(&a, &b);
    temp = *a;                 printf("a=%d i b=%d", a, b);
    *a = *b;
    *b = temp;
}
```

Одредити које вредности ће променљиве `a` и `b` добити после извршавања кода. Изабрати један од понуђених одговора:

1. a=3 i b=3
2. a=4 i b=4
3. a=3 i b=4
4. a=4 i b=3

51. Потребно је у програмском језику Ц дефинисати функцију **pravougaonik()** која израчунава обим и површину правоугаоника. Функција **pravougaonik()** преко вредности функције враћа обим правоугаоника, а преносом параметара по референци и показивача **poc** враћа површину правоугаоника. Улазни аргументи функције су странице **a** и **b** које су типа float. Анализирати дате дефиниције функције и проценити која од њих је исправна.

Заокружити број испред очекиваног одговора:

```
1. float pravougaonik(float a, float b, float *poc) {
    return 2*a+2*b;
    *poc=a*b;
}
2. float pravougaonik(float a, float b, float poc) {
    poc=a*b;
    return 2*a+2*b;
}
3. float pravougaonik(float a, float b, float &poc) {
    poc=a*b;
    return 2*a+2*b;
}
4. float pravougaonik(float a, float b, float *poc) {
    *poc=a*b;
    return 2*a+2*b;
}
```

2

52. Дат је код рекурзивне функције написан у програмском језику Ц:

```
void prikaz(int k, int n) {
    printf("%d\t", k);
    if(k<n) prikaz(k+1,n);
    printf("%d\t", k);
}
```

Проценити шта ће се десити ако се функција позове наредбом: `prikaz(4,10);`

2

Заокружити број испред тачног одговора:

1. 4 5 6 7 8 9 10
2. 4 5 6 7 8 9 10 9 8 7 6 5 4
3. 4 5 6 7 8 9 10 10 9 8 7 6 5 4
4. 10 9 8 7 6 5 4

53. Дат је код рекурзивне функције написан у програмском језику Ц. Проценити шта ће се десити након њеног извршавања.

```
int fun(int a[], int n)
{
    return (n>0) ? a[0] + fun(a+1, n-1) : 0;
}
```

Заокружити број испред тачног одговора:

2

1. Сортира низ **a** од **n** елемената у растућем редоследу.
2. Сваки елемент низа **a** од **n** елемената инкрементује
3. Сваки елемент низа **a** од **n** елемената увећава за вредност првог елемента низа **a**
4. Израчунава суму елемената низа **a** од **n** елемената.
5. Обрће елементе низа **a** од **n** елемената.

54. У програмском језику Ц дат је прототип функције **funkcija()** и декларисане су променљиве у функцији **main()**. У понуђеним одговорима дати су позиви функције за декларисане променљиве.

```
void funkcija(int *x, int *y, int **p);
void main(){
    int a=5, b=7, c=15, *poc;
    poc = &c;
}
```

2

Заокружити редни број испред исправно записаног позива декларисане функције:

1. funkcija(a, b, &poc);
2. funkcija(&a, &b, &poc);
3. funkcija(&a, &b, poc);
4. c = funkcija(&a, &b, &poc);

55. Дат је део кода програма написаног у програмском језику Ц. Одредити шта ће се исписати на екрану након његовог извршавања.

```
char s[20]={ 'A', 'c', 'a', ' ', 'j', 'e', '\0', 'd', 'o', 'b', 'a', 'r' };
char t[12]={ 'A', 'c', 'a', ' ', 'j', 'e', '\0', 'd', 'o', 'b', 'a', 'r' };
char *poc=s+7;

printf("\nString = %s\n",poc);
printf("String = %s\n",s+4);
printf("Znak = %c\n",*poc);
printf("String = %s\n",t+7);
```

2

Заокружити број испред очекиваног одговора:

- | | |
|--|---|
| 1. String = Aca je dobar
String = Aca
Znak = A
Aca je dobar | 2. String = Aca
String = je
Znak = d
Neće prikazati ništa |
| 3. String = dobar
String = je
Znak = d
Nepredvidivo jer string t ima 12 znakova
koliko i rezervisani prostor, a poslednji
znak nije \0' | 4. String = dobar
String = dobar
Znak = d
Nepredvidivo jer string t ima 12 znakova
koliko i rezervisani prostor, a poslednji
znak nije \0' |

56. Дат је код функције **fun()** написане у програмском језику Ц. Изабрати којој функцији из стандардне библиотеке функција **ctype.h** одговара дата функција.

```
int fun(char c)
{
    return ((c>='a' && c<='z') || 
            (c>='A' && c<='Z') || 
            (c>='0' && c<='9')) ? 1 : 0;
}
```

1

Заокружити број испред тачног одговора:

1. isupper
2. isalpha
3. gets
4. strncat
5. atoi
6. strchr
7. strcmp
8. isalnum

57. Дат је код функције **fun()** написане у програмском језику Ц. Изабрати којој функцији из стандардне библиотеке функција **ctype.h** одговара дата функција.

```
int fun(char c)
{
    return (c>='A' && c<='Z') ? 1 : 0;
}
```

Заокружити број испред тачног одговора:

1. isupper
2. isalpha
3. gets
4. strncat
5. atoi
6. strchr
7. strcmp

1

58. Дат је код функције **fun()** написане у програмском језику Ц. Изабрати којој стандардној функцији одговара дата функција.

```
char *fun(char *s) {
    char c,*temp;
    temp=s;
    while((c=getchar()) != '\n') *temp++=c;
    *temp='\0';
    return s;
}
```

Заокружити број испред тачног одговора:

1

1. isupper
2. isalpha
3. gets
4. strncat
5. atoi
6. strchr
7. strcmp

59. Дат је код функције **fun(...)** написане у програмском језику Ц. Изабрати којој стандардној функцији одговара дата функција.

```
char *fun(char *s, char *t, int n) {
    char *temp=s;
    while(*s) s++;
    while(*t && n--) *s++=*t++;
    *s='\0';
    return temp;
}
```

Заокружити број испред очекиваног одговора:

2

1. isupper
2. isalpha
3. gets
4. strncat
5. atoi
6. strchr
7. strcmp

60. У програмском језику Ц декларисане су променљиве и дат је део кода програма:

```
FILE *fp;
char str[150];
fp=fopen("primer.txt", "r");
fgets(str, 80, fp);
```

Одредити шта је последица извршавања датог кода. Заокружити број испред очекиваног одговора:

1. Учитава 80 карактера из датотеке и смешта у стринг str.
2. Учитава максимално 150 карактера из датотеке и смешта у стринг str
3. Учитава стринг из датотеке све док се не учита знак за нови ред или 80 карактера
4. Учитава стринг из датотеке све док се не учита знак за нови ред или 150 карактера

1

61. Дат је код функције **fun(...)** написане у програмском језику Ц. Изабрати којој стандардној функцији одговара дата функција.

```
int fun(char *s) {
    int n, sign;
    while (*s==' ' || *s=='\t') s++;
    sign = (*s=='-') ? -1 : 1;
    if (*s=='+' || *s=='-') s++;
    for (n=0; *s>='0' && *s<='9'; s++) n=10*n+ *s - '0';
    return (!*s) ? sign*n : 0;
}
```

Заокружити број испред очекиваног одговора:

2

1. isupper
2. isalpha
3. gets
4. strncat
5. atoi
6. strchr
7. strcmp

62. Дат је код функције **fun (...)** написане у програмском језику Ц. Анализирати код и одредити којој стандардној функцији одговара дата функција.

```
char *fun(char *s, char c)
{
    while (*s && *s!=c) s++;
    return (*s) ? s : NULL;
}
```

Заокружити број испред очекиваног одговора:

2

1. isupper
2. isalpha
3. gets
4. strncat
5. atoi
6. strchr
7. strcmp

63. Дат је код функције **fun(...)** написане у програмском језику Ц. Анализирати код и одредити којој стандардној функцији одговара дата функција.

```
int fun(char *s, char *t) {
    char tempt, temps;
    while(*s && *t) {
        if(*t>='A' && *t<='Z') tempt = 'a' + *t - 'A';
        else tempt= *t;
        if(*s>='A' && *s<='Z') temps = 'a' + *s - 'A';
        else temps= *s;
        if(temps != tempt) return temps - tempt;
        else s++, t++;
    }
    return *s - *t; .
}
```

2

Заокружити број испред очекиваног одговора:

1. isupper
2. isalpha
3. gets
4. strncat
5. atoi
6. strchr
7. strcmp

64. У програмском језику Ц декларисане су променљиве и дат је део кода програма:

```
FILE *fp;
char str[150]; int br=10;
fp=fopen("primer.txt", "r");
fseek(fp, br, SEEK_CUR);
fgets(str, 80, fp);
```

1

Одредити шта је последица извршавања датог кода. Заокружити број испред очекиваног одговора:

1. Учитава 80 карактера из датотеке и смешта у стринг str.
2. Учитава максимално 150 карактера из датотеке и смешта у стринг str. Ако у датотеци нема 150 карактера, учита br=10 карактера.
3. Учитава стринг из датотеке све док се не учита знак за нови ред или 80 карактера, и то почев од 11. карактера у односу на почетак датотеке.
4. Учитава стринг из датотеке све док се не учита знак за нови ред или 150 карактера, почев од 80.-тог карактера у датотеци.

65. Дат је део кода у програмском језику Ц, који омогућава отварање датотеке **PRIMER.TXT** са модом приступа за писање. Ако тражена датотека не постоји формирати нову празну датотеку са тим именом. У датотеку уписати реч **PROGRAMIRANJE**. У коду програма недостаје наредба у првој линији. Допунити код једном од понуђених наредби.

1. _____
2. fp = fopen("primer.txt", "w");
3. fputs("PROGRAMIRANJE", fp);
4. fclose(fp);

1

Заокружити број испред тачне наредбе:

1. FILE fp;
2. FILE *fp;
3. FILE &fp;
4. FILE *(fp);

66. У програму написаном у програмском језику Ц декларисана је променљива **pod** типа **int**. Употребом функције **fprintf(...)** уписати декларисан податак у стандардну излазну датотеку.

Заокружити број испред исправно написане наредбе:

1. `fprintf(pod);`
2. `fprintf("%d", pod);`
3. `fprintf("%d", pod, stdin);`
4. `fprintf(stdout, "%d", pod);`

1

67. Дат је следећи код написан у програмском језику Ц:

```
FILE *fp;
if((fp = fopen("ulaz.txt","r")) == NULL) {
    printf("Datoteka ulaz.txt ne moze biti otvorena!!!");
    return;
}
fclose(fp);
```

Анализирати дати код и заокружити број испред одговора који описује шта ће се десити након његовог извршавања:

2

1. Отварање бинарне датотеке за читање, ако датотека није успешно отворена приказаће се порука : "Datoteka ulaz.txt ne moze biti otvorena!!"
2. Отварање текст датотеке за читање, ако датотека није успешно отворена приказаће се порука: "Datoteka ulaz.txt ne moze biti otvorena!!"
3. Отварање текст датотеке за писање, ако датотека није успешно отворена приказаће се порука: "Datoteka ulaz.txt ne moze biti otvorena!!"
4. Креирање и отварање текст датотеке за упис, ако датотека није успешно отворена приказаће се порука: "Datoteka ulaz.txt ne moze biti otvorena!!"

68. У програмском језику Ц декларисан је структурни тип података **Ucenik**, а затим и променљива која представља низ од max 30 ученика:

```
typedef struct
{
    char ime[50];
    int razred;
    int ocene[10];
}Ucenik; ...
Ucenik T[30];
```

1

Заокруживањем обележити наредбу којом се првом ученику у низу **T** уписује оцена 5 из математике, ако знамо да је математика четврта оцена у низу оцена:

1. `T[0].ocene[3]='5';`
2. `T[1].ocene[4]=5;`
3. `T[0].ocene[3]=5;`
4. `T.ocene[3]=5;`

69. У програмском језику Ц декларисана је променљива **fp**: `FILE *fp;`

и дата је линија кода програма `int poz=fteell(fp);`

Одредити шта је последица извршавања дате линије кода. Заокружити број испред очекиваног одговора:

1

1. Позиционира се на почетак датотеке
2. Позиционира се на крај датотеке
3. Затвара датотеку
4. Утврђује тренутну позицију у датотеци

70. У програмском језику Ц декларисан је структурни тип података ***Ucenik***, а затим и променљива која представља низ од max 30 ученика:

```
typedef struct          typedef struct
{
    char naziv[30];      {
    int razred, ocena;   char ime[20], prezime[20];
} Predmet;              Predmet predmeti[10];
                        }Ucenik;
...
int i; Ucenik T[30];
```

1

Заокруживањем обележити наредбу којом ће се првом ученику у низу **T** уписати оцена 5 из математике, ако знамо да је математика четврти предмет у низу предмета:

1. **T[0] ->predmeti[3] ->ocena=5;**
2. **T[0].predmeti[3].ocena=5;**
3. **T[0].predmeti["matematika"].ocena=5;**
4. **T.ocena[3]=5;**

71. У програмском језику Ц декларисани су структурни типови података ***Tacka*** (одређена координатама), ***Poligon*** (одређен бројем и координатама темена) и ***Piramida*** (одређена типом основе – троугао, четвороуга... и висином). Потом је декларисана и променљива типа ***Piramida**:

```
typedef struct          typedef struct          typedef struct
{
    float x, y;          {
    int brojTemeni;      Tacka temena[10];
} Tacka;                } Poligon;           Poligon osnova;
                        } Poligon;           float visina;
                        } Piramida;
```

Piramida *p;

2

Заокружити број испред наредбе којом се број темена основе пирамиде на коју показује декларисани показивач ***p**, поставља на 6:

1. **p.osnova.brojTemeni=6;**
2. **p.osnova->brojTemeni=6;**
3. **p->osnova.brojTemeni=6;**
4. **p->osnova[brojTemeni]=6;**
5. **p->osnova->brojTemeni=6;**

72. У програмском језику Ц декларисани су структурни типови података ***Tacka3D*** (који дефинише тачку у простору) и ***Lopta*** (одређена центром и полуупречником):

```
typedef struct          typedef struct
{
    float x, y, z;      {
    Tacka3D centar;
} Tacka3D;              float R;
                        } Lopta;
```

1

Заокружити број испред исправно написане наредбе декларације и иницијализације променљиве **x** типа ***Lopta***, тако да јој центар буде у тачки O(2,2,2), а полуупречник 10цм:

1. **Lopta x={10, {2, 2, 2}};**
2. **Lopta x={2, 2, 2, 10};**
3. **Lopta x={2, 2, 2}, {10};**
4. **Lopta x={{2, 2, 2}, 10};**

73. У програмском језику Ц је декларисани су структурни типови података **Datum** и **Letovanje**:

```
typedef struct { int dan, mesec, godina; } Datum;
typedef struct { char destinacija[50]; Datum polazak, povratak; float cena; } Letovanje;
```

Заокружити број испред исправно написане наредбе декларације и иницијализације променљиве **x** типа **Letovanje**, ако је дестинација Родос, полазак 10.08.2016., а повратак 21.08.2016. Цена је 480€:

1. Letovanje x={"Rodos", {2016, 8, 10}, {2016, 8, 21}, 480};
2. Letovanje x={"Rodos", {10, 8, 2016}, {21, 8, 2016}, 480};
3. Letovanje x={"Rodos", {10, 8, 2016, 21, 8, 2016}, 480};
4. Letovanje x={Rodos, {2016, 8, 10}, {2016, 8, 21}, 480};
5. Letovanje x={"Rodos", 480, {2016, 8, 10}, {2016, 8, 21}};

2

74. У програмском језику Ц је декларисани су структурни типови података **Zaposleni** и **Odeljenje**, а затим и променљива типа **Odeljenje**:

```
typedef struct { char ime[50], prezime[50]; float zarade[12]; //plata po mesecima } Zaposleni;
typedef struct { char naziv[50]; Zaposleni radnici[200]; //niz svih zaposlenih } Odeljenje;
...
Odeljenje marketing;
```

2

Заокружити број испред наредбе којом се раднику, који се у евиденцији одељења **marketing** налази на првој позицији, уписује плату за децембар у износу од 70.000дин:

1. marketing.radnici[1].zarade[12]=70000;
2. marketing[0].radnici[0].zarade[11]=70000;
3. marketing.radnici[0].zarade[11]=70000;
4. marketing[0].radnici.zarade[11]=70000;

75. У програмском језику Ц је декларисан је структурни тип података **Ugao**, а затим и декларација функције **PrviVeciOd** која у скупу углова запамћених у улазном фајлу, одређује први угао већи од неког задатог угла.

```
typedef struct { int stepen, minut, sekund; } Ugao;
Ugao PrviVeciOd(char ime[], Ugao referentni) {...}
```

Ако су у позивајућој функцији декларисане променљиве:

```
Ugao alfa, beta; char name[20]; FILE *in;
```

2

Заокружити број испред коректно написане наредбе позива функције:

1. beta=PrviVeciOd(in, alfa);
2. beta=PrviVeciOd(name, alfa)
3. beta=PrviVeciOd(name[20], alfa)
4. PrviVeciOd(in, alfa, *beta)

76. У програмском језику Ц декларисана је променљива fp: **FILE *fp;** и дата је линија кода програма **fseek(fp,br,SEEK_CUR);**

Одредити шта је последица извршавања дате линије кода. Заокружити број испред очекиваног одговора:

1. Позиционира се на почетак датотеке
2. Позиционира се на крај датотеке
3. Позиционира се на br бајтова од почетка датотеке
4. Позиционира се на br бајтова од краја датотеке
5. Позиционира се на br бајтова од тренутне позиције у датотеци

1

77. У програмском језику Ц дат је код програма у коме заглавље **main** има следећи изглед:

```
void main(int argc, char *argv[]).
```

Програм PRIMER.EXE који има овакво заглавље main функције, стартовати из командне линије наредбом:

```
C:\TC\BIN>PRIMER A:\primer1.txt A:\primer2.txt
```

Одредити које су вредности улазних аргумената argc и argv функције main након стартовања програма PRIMER.EXE. Заокружити број испред тачног одговора:

2

1. argc = 3 argv[0] = PRIMER argv[1] = A:\primer1.txt argv[2] = A:\primer2.txt
2. argc = 2 argv[0] = A:\primer1.txt argv[1] = A:\primer2.txt
3. argc = 2 argv[0] = primer1.txt argv[1] = primer2.txt
4. argc = 3 argv[0] = PRIMER argv[1] = primer1.txt argv[2] = primer2.txt

78. У програму написаном у програмском језику Ц декларисана је променљива fp која представља показивач на бинарну датотеку и променљива **podatak** у коју ће се уписати прочитани подаци из дефинисане бинарне датотеке.

Заокружити редни број испред наредбе која омогућава учитавање три бајта са тренутне позиције из бинарне датотеке:

1. fread(podatak, 24, 1, fp);
2. fread(&podatak, 24, 0, fp);
3. fread(&podatak, 3, 1, fp);
4. fscanf(&podatak, 3, 1, fp);
5. fscanf(fp, 3, &podatak);

1

79. У програму написаном у програмском језику Ц декларисана је променљива fp која представља показивач на бинарну датотеку и променљива **podatak** чија вредност ће се уписати у дефинисану бинарну датотеку.

Заокружити редни број испред наредбе која омогућава упис три бајта у бинарну датотеку:

1. fprintf(&podatak, 3, 1, fp);
2. fprintf(fp, 3, &podatak);
3. fwrite(podatak, 24, 1, fp);
4. fwrite(&podatak, 24, 0, fp);
5. fwrite(&podatak, 3, 1, fp);

1

У следећим задацима заокружите бројеве испред тражених одговора

<p>80. Наведене су тврдње које се односе на превођење кода процесом компилације. Заокружити бројеве испред ТАЧНИХ тврдњи:</p> <ol style="list-style-type: none"> 1. Код компилаторских језика, после сваке промене у извornом коду неопходно је поновити процес превођења. 2. Код превођења компилацијом приликом извршења програма, у РАМ меморији рачунара се МОРАЈУ налазити и изворни и извршни код 3. Компилатор преводи и извршава ред по ред извornог кода. 4. Код компилације програма процес превођења и процес извршења су временски независни. 	2
<p>81. Дате су наредбе декларације променљивих (са и без иницијализације вредности) написане на програмском језику Ц. Заокружити бројеве испред исправно написаних наредби декларације променљивих:</p> <ol style="list-style-type: none"> 1. <code>int a=b=c=5;</code> 2. <code>int a=5, b=5, c=5;</code> 3. <code>char zn="a";</code> 4. <code>long a; b=5; c;</code> 5. <code>int a=0xf2;</code> 6. <code>char zn='\\b';</code> 	1,5
<p>82. Дата је наредба декларације int a, b; Имајући у виду дату декларацију, заокружити бројеве испред НЕИСПРАВНО написаних наредби форматираног уноса података:</p> <ol style="list-style-type: none"> 1. <code>scanf("%d%f", &a, &b);</code> 2. <code>scanf("%d%*d", &a, &b);</code> 3. <code>scanf("%d%d", &a, &b);</code> 4. <code>scanf("%d%d", a, b);</code> 5. <code>scanf("%d%*d", &a);</code> 6. <code>scanf("%5d%5d", &a, &b);</code> 	1,5
<p>83. Декларисане су следеће променљиве:</p> <pre style="margin-left: 40px;"><code>float x, z; const float y;</code></pre> <p>Заокружити бројеве испред НЕИСПРАВНО написаних наредби доделе вредности променљивама:</p> <ol style="list-style-type: none"> 1. <code>x %= y;</code> 2. <code>x =+ 5;</code> 3. <code>x += y + 5;</code> 4. <code>x =/ y + 5;</code> 5. <code>y = x + z;</code> 6. <code>x = z = y + 5;</code> 	1,5

84. Дата је наредба у Ц језику, која температуру у Целзијусима **tempc** претвара у температуру у Фаренхайтима **tempf**. Подаци tempc и tempf су реални бројеви обичне тачности. Проценити који изрази дају тачно решење.

1. $\text{temf} = (9 / 5) * \text{tempc} + 32;$
2. $\text{temf} = 9 / 5 * \text{tempc} + 32;$
3. $\text{temf} = 9 * \text{tempc} / 5 + 32;$
4. $\text{temf} = 32 + 9 * \text{tempc} / 5;$

2

85. Дата је if-else наредба:

```
if (a==3 || a==5) p++;
else p--;
```

Заокружити бројеве испред кода који представља дату структуру приказану помоћу једне switch наредбе:

1. `switch(a) {
 case 3: p++; break;
 case 5: p++; break;
 default: p--;
 }`
2. `switch(a) {
 case 3: case 5: p++; break;
 p--;
 }`
3. `switch(a) {
 case 3: case 5: p++; break;
 default: p--;
 }`
4. `switch(a) {
 case 3: case 5: p++;
 default: p--;
 }`

2

86. Наведени су искази који се односе на дефиницију while циклуса.

Заокружити бројеве испред **ТАЧНИХ** исказа:

1. while циклус се извршава све док је услов логичка неистина (једнак нули),
2. while циклус се користи када се зна колико ће се пута циклус извршавати,
3. у while циклусу се увек прво проверава да ли је услов логичка истина, те ако јесте наредба се извршава
4. код while циклуса се може десити да се тело циклуса не изврши ниједном (на почетку услов није задовољен).

1

87. Наведени су искази који се односе на дефиницију do while циклуса.

Заокружити бројеве испред **ТАЧНИХ** исказа:

1. Користи се када се не зна колико ће се пута циклус понављати,
2. Прво се извршава тело циклуса, а затим израчунава вредност логичког израза. Ако се добије логичка неистина, циклус се поновно извршава.
3. Циклус се завршава када услов добија вредност логичке истине
4. Циклус се извршава барем једном.

1

88.	<p>Дате су наредбе декларације и иницијализације једнодимензионалног низа целих бројева у програмском језику Ц. Заокружити бројеве испред исправно написаних наредби декларације и иницијализације једнодимензионалног низа:</p> <ol style="list-style-type: none"> 1. <code>int a[10]={1,2,3};</code> 2. <code>int a[5]={-3, -2, -1, 0, 1, 2, 3};</code> 3. <code>int a[]={10,20,30,40,50};</code> 4. <code>int[5] a={1, 2, 3, 4, 5};</code> 5. <code>int a={10,20,30,40,50};</code> 6. <code>int a[5]={'1', '2', '3', '4', '5'};</code> 	1,5
89.	<p>У програму на програмском језику Ц извршена је следећа декларација, а касније и резервација меморијског простора за низ реалних бројева обичне талности дужине n:</p> <pre style="font-family: monospace; margin: 0;"><code>float *B; int n; B=calloc(n, sizeof(float));</code></pre> <p>Означи исправно написане наредбе за ПРИКАЗ i-тог елемента низа B:</p>	2
90.	<p>У програму на програмском језику Ц извршена је следећа декларација, а касније и резервација меморијског простора за низ реалних бројева обичне талности дужине n:</p> <pre style="font-family: monospace; margin: 0;"><code>float *B; int n; B=calloc(n, sizeof(float));</code></pre> <p>Означи исправно написане наредбе за УНОС i-тог елемента низа B:</p>	2
91.	<p>Термин „адресна аритметика“ се односи на извођење аритметичких операција над показивачима. Анализирати дате исказе који дефинишу дозвољене аритметичке операције над показивачима. Заокружити бројеве испред тачних исказа:</p> <ol style="list-style-type: none"> 1. Додела вредности једног показивача другом. 2. Додавање рационалног податка на вредност показивача и одузимање рационалног податка од вредности показивача. 3. Одузимање и упоређивање два показивача. 4. Идентификатор низа је показивач на почетак низа и може му се мењати вредност. 5. Упоређивање показивача са NULL. 6. Сабирањем два показивача, добија се нови показивач. 	1,5

92. У програмском језику Ц дата је наредба која декларише променљиве **a, b и c.**

```
struct elem1{
    int x,y;
};

struct elem{
    struct elem1 poz;
    int oblik;
};

struct elem a = {{1,5},3}, *b, c;
```

Заокружити број испред одговора у ком су наредбе доделе између декларисаних променљивих **НЕТАЧНО** написане:

- | | |
|--------------------------|---------------------------|
| 1. c.poz.x = a.poz.x; | 2. b->poz.x = a.poz.x; |
| c.poz.y = a.poz.y; | b->poz.y = a.poz.y; |
| c.oblik = a.oblik; | b->oblik = a.oblik; |
| 3. c = a; | 4. b=a; |
| 5. b=&a; | 6. *b=a; |

3

93. У програмском језику Ц написати функцију **xalloc** која додељује меморијски простор вектору од **n** елемената и врши иницијализацију његових елемената. Сваки елемент вектора заузима меморијски простор величине **size**. У седму линију кода додати наредбу која недостаје да би функција извршавала свој задатак.

```
1. void *xalloc(int n, int size)
2. {
3.     int i;
4.     char *p, *q;
5.
6.     if((p = q = malloc(n*size)) != NULL) {
7.         _____
8.     }
9.     return q;
10. }
```

2

Заокружити бројеве испред наредби које представљају могућа решења овог проблема:

1. **for**(i=0; i<n*size; i++) *p++=0;
2. **for**(i=0; i<size; i++) p++=0;
3. **for**(i=0; i<n*size; i++) *p=0;
4. **for**(i=0; i<n*size; *p++=0, i++) ;

94. Наредбама програмског језика Ц дата је декларација, а касније у коду и иницијализација динамичке матрице:

```
int **mat;
```

Заокружити бројеве испред понуђених могућих начина обележавања вредности елемента који се налази у i-тој врсти и j-тој колони динамичке матрице **mat**:

1. *(mat[i] + j)
2. *(mat + i + j)
3. * mat[i] + j
4. *(*(mat + i) + j)
5. **(mat + i + j)
6. mat[i][j]

1,5

95. Наредбама програмског језика Ц, дата је декларација једне симболичке константе и једне константне променљиве:

```
#define k 50 ...
int m=100; ...
```

Заокружити бројеве испред исправно написаних наредби декларације дводимензионалног низа целих бројева (матрице):

1. int a [k][k];
2. int b [k][m];
3. int c [k][10];
4. int x [100][50];
5. int y [10, 10];
6. int z [m][10];

1,5

96. Дати су прототипови функција написани у програмском језику Ц. Заокружити бројеве испред исправно написаних прототипова функција:

1. float* pp1(int a, int b, int c);
2. int pp2(int a[] [10], int n);
3. int pp3(int a[], n; float b);
4. void pp4(int *a, int n);
5. int pp5(int a[] [], int n);
6. int pp6(int a[], int n);
7. int pp7(int a, b, c);
8. float[] pp8(float a[], int n);

2

97. Дат је прототип функције написан у програмском језику Ц:

```
void Saberi(int n, int *a, int *b);
```

У main функцији дате су следеће декларације променљивих:

```
int x[50][50], y[50], m, j, i;
```

Заокружити бројеве испред исправно написаних позива декларисане функције:

1. Saberi(m, y[i], y[i+1]);
2. Saberi(y[i], x[i], x[i+1]);
3. Saberi(m, y, x[i][j]);
4. Saberi(y, x[i], x[i+1]);
5. Saberi(10, y, x[0]);
6. Saberi(x[i][j], x[i], x[j]);

3

98. Имајући у виду декларације дате на програмском језику Ц:

```
char s1[100] = "xyzw", s2[100], *s3 = "xyzw", *s4;
```

Заокружи наредбе које ће изазвати грешку приликом извршења (run-time error):

1. strcpy(s1, "maturski ispit");
2. strncpy(s1, "maturski ispit", 8);
3. strcpy(s2, "maturski ispit");
4. strcpy(s3, "maturski ispit");
5. strncpy(s3, "maturski ispit", 3);
6. strcpy(s4, "maturski ispit");

2

99. Дат је прототип функције написан у програмском језику Ц:

```
int Zbir(int n, int m, float **a);
```

У main функцији дате су следеће декларације променљивих:

```
float M[20][50], *V[20], W;  
int i, j;
```

Имајући у виду прототип функције, заокружити бројеве испред наведених стварних параметара које се приликом позивања функције **Zbir** могу наћи на месту фиктивног параметра **a**:

1. V[i]
2. M[i]
3. M[i][j]
4. W
5. M
6. V

2

100. Дат је прототип функције написан у програмском језику Ц:

```
void Umetni(char *a, char k);
```

У main функцији дате су следеће декларације променљивих:

```
char s1[20], *s2, s3;
```

Заокружити бројеве испред исправно написаних позива декларисане функције:

3

1. Umetni(s2, s1[i]);
2. Umetni(s2, s1);
3. Umetni(s2, 'A');
4. Umetni(s1, s3);
5. Umetni(*s2, s3);
6. Umetni(s3, &s1);

101. У програмском језику Ц, на основу начина смештања података у датотеку, разликују се две врсте датотека. Заокружи редне бројеве испред побројаних типова датотека који се односе на поменуту класификацију:

1. Индекс-секвенцијална датотека
2. Текстуална датотека
3. Бинарна датотека
4. Zip датотека

1

102. Заокружити бројеве испред понуђених тврдњи које представљају тачне наставке изјаве која се односе на повратну вредност функције **fopen**:

При покушају да датотеку отворимо за писање, функција fopen...

1. ако датотека не постоји, креира празну датотеку, поставља се на њен почетак и враћа показивач на ту датотеку
2. враћа NULL показивач ако датотека не постоји
3. ако датотека постоји, излази упозорење да ће њен садржај бити уништен при отварању
4. ако датотека не постоји, изазива грешку која доводи до пуцања програма
5. ако датотека постоји, уништава њен садржај без упозорења

2

103. Заокружити бројеве испред понуђених тврдњи које представљају тачне наставке изјаве која се односе на повратну вредност функције `fopen`:

При покушају да датотеку отворимо за читање, функција fopen...

1. ако датотека не постоји, изазива грешку која доводи до пуцања програма
2. ако датотека не постоји, креира празну датотеку, поставља се на њен почетак и враћа показивач на ту датотеку
3. враћа NULL показивач ако датотека не постоји
4. ако датотека постоји, враћа показивач на ту датотеку

1

104. У програмском језику Ц је декларисана низовна променљива:

```
int niz[10];
```

Заокружити бројеве испред исправно написаних наредби читања низа целих бројева дужине 10 из бинарног фајла на који показује показивач `*in`:

1. `fread(niz, sizeof (int), 10, in);`
2. `fread(&niz, sizeof (int), 10, in);`
3. `fread(&niz, sizeof niz, 1, in);`
4. `fread(niz, sizeof niz, 1, in);`
5. `fread(niz, sizeof (niz), 1, *in);`
6. `fread(niz, sizeof (int)*10, in);`

2

105. У програмском језику Ц декларисан је структурни тип података ***Ucenik***, а затим и променљива типа ***Ucenik***:

```
typedef struct
{
    char ime[50];
    int razred;
    int ocene[10];
}Ucenik; ...

int i; Ucenik x;
```

2

Заокруживањем обележити исправне начине приступа пољима структурне променљиве ***x***:

1. `x.ocene[i]`
2. `*x.razred`
3. `x->ime`
4. `x[i].ocene`
5. `x.ime`

106. У програмском језику Ц декларисан је структурни тип података ***Putovanje***, а затим и променљива типа ****Putovanje***:

```
typedef struct
{
    char start[50], cilj[50];
    int kilometraza;
}Putovanje; ...

Putovanje *p;
```

2

Заокруживањем обележити исправне начине приступа пољима структурне променљиве:

1. `*p->kilometraza`
2. `(*p).kilometraza`
3. `&p->kilometraza`
4. `p->start`
5. `* (p).start`

107. У програмском језику Ц дефинисане су структуре које омогућавају груписање података различитих типова. Дефинисати структурни тип података **Tucenik** са пољима **ime** (максимално 30 карактера), **prezime** (максимално 30 карактера) и **prosek** (типа double).

Заокружити бројеве испред исправно написаних декларација структурног типа **Tucenik**:

1. <code>typedef struct ucenik{ char ime[30]; char prezime[30]; double prosek; }Tucenik;</code>	2. <code>typedef struct ucenik{ char ime[31]; char prezime[31]; double prosek; }Tucenik;</code>
3. <code>struct ucenik{ char ime[30]; char prezime[30]; double prosek; }Tucenik;</code>	4. <code>typedef struct ucenik{ char ime[31],prezime[31]; double prosek; }Tucenik;</code>

1

2

108. У програмском језику Ц дат је код програма који формира бинарну датотеку **ULAZ.DAT** и декларише податак структурног типа **Tucenik a**, затим га иницијализује следећим вредностима: **Petrovic Vukasin 5.00**. Допунити програмски код (линија 11) наредбом помоћу које се у отвореном бинарном фајлу врши упис формираног податка **a** типа **Tucenik**.

```

1. #include <stdio.h>
2. typedef struct ucenik{
3.     char ime[31];
4.     char prezime[31];
5.     double prosek;
6. }Tucenik;
7. int main(){
8.     FILE *fp;
9.     Tucenik a={"Vukasin", "Petrovic", 5.00};
10.    fp = fopen("ULAZ.DAT", "wb");
11.    _____
12.    fclose(fp);
13.    return 0;
14. }
```

Заокружити бројеве испред тачних одговора:

1. `fwrite(&a, sizeof(Tucenik), 1, fp);`
2. `fwrite(a, sizeof(Tucenik), 1, fp);`
3. `fwrite(&a, sizeof a, 1, fp);`
4. `fwrite(fp, sizeof(Tucenik), 1, a);`

Допуните следеће реченице и табеле

109. Дата је наредба форматираног уноса вредности у променљиве, написана на програмском језику Ц: `scanf("%o%c%e", &x, &y, &z, &w);`
За сваку променљиву, на основу употребљене конверзије, одреди ког је она типа и упиши на одговарајућу линију:

1. променљива **x** је типа: _____
2. променљива **y** је типа: _____
3. променљива **z** је типа: _____
4. променљива **w** је типа: _____

2

110. Дата су наредба декларације, а затим и наредба форматираног уноса вредности у променљиве, написана на програмском језику Ц:

```
int x, y;
scanf("%3i%3i", &x, &y);
```

Следи тастатурни унос у облику: 12345 12345

2

За сваку променљиву одредити и на одговарајућу линију уписати, коју ће вредност променљива имати по извршењу наредби:

1. променљива **x** добија вредност **x =** _____
2. променљива **y** добија вредност **y =** _____

111. Дата су наредба декларације, а затим и наредба форматираног уноса вредности у променљиве, написана на програмском језику Ц:

```
int a, b;
scanf("%d*c%d", &a, &b);
```

Следи тастатурни унос у облику: 2.3 5

2

За сваку променљиву одредити и на одговарајућу линију уписати, коју ће вредност променљива имати по извршењу наредби:

1. променљива **a** добија вредност **a =** _____
2. променљива **b** добија вредност **b =** _____

112. Дати је декларација променљивих `int a=3, b=15;`

Израчунати вредност коју ће променљиве имати по извршењу следеће наредбе:

b %= ++ **a**;

2

a = _____
b = _____

113. Дати су изрази формирани коришћењем математичких оператора. Водећи рачуна о типовима података, одредити вредности датих израза и уписати их линију у продужетку. Ако израз изазива грешку, уместо вредности, написати **error**:

1. $10 / 4. =$ _____
2. $10. / 5 =$ _____
3. $-10 \% 3 =$ _____
4. $10. \% 5 =$ _____
5. $10 \% (-3) =$ _____
6. $(100/3) \% 6 =$ _____

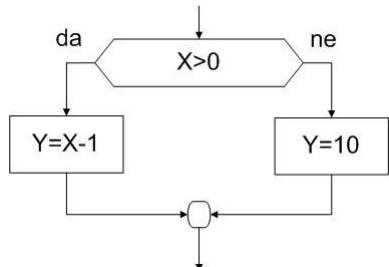
3

114. Допунити реченицу:

Оператором инкрементирања се врши _____ целобројне променљиве за вредност _____.

1

115. Наредбу гранања представљену алгоритмом на слици написати коришћењем тернарног оператора:



2

116. Синтаксом програмског језика Ц написати следећи израз:

$$z = \sqrt{x^{2*n} + y^{2*m}}$$

1

117. Одредити вредности које ће променљиве x и y имати по извршењу следећег кода:

```

int x=10;
int y=20;
if(x>50)
    x-=10;
    y+=10;
  
```

2

Уписати добијене вредности на предвиђене линије:

x = _____ y = _____

118. У програмском језику Ц декларисане су две целобројне променљиве:

```
int x=0, izbor;
```

За дате вредности промељиве **izbor**, одреди вредност промељиве **x** по извршењу следеће наредбе вишеструког гранања и уписати их на предвиђене линије:

```

switch(izbor)
{
    case 1: x += 1;
    case 2: x += 2; break;
    case 3: x += 3;
    default: x = 100;
    case 4: x += 4;
    case 5: x += 5;
}
  
```

2

1. за izbor=3, x=_____

3. за izbor=4, x=_____

2. за izbor=10, x=_____

4. за izbor=2, x=_____

119. Дата је декларација једнодиманзионалног низа и део програмског кода:

```
int p[200], i, n=10;
for(i=0; i<n; i++) scanf("%d", &p[i]);
```

Одредити вредности израза и уписати их на одговарајућу линију:

`sizeof p` = _____
`sizeof p[0]` = _____

2

120. Дате су следеће декларације: `int p[200], i, n, pom;`

А затим и део кода који треба да врши циклично померање елемената низа `p` дужине `n`, за једно место **удесно**. У коду недостају неки од елемената.

Имајући у виду дату иницијализацију петље, у предвиђена поља унеси одговарајуће елементе **преписивањем израза** из листе понуђених израза (подразумевати да су све потребне променљиве иницијализоване):

```
pom = _____;
for(i=n-2; i _____; _____)
    _____ = _____;
    _____ = pom;
```

1. `p[0]`
2. `p[n-1]`
3. `p[n]`
4. `p[i+1]`
5. `p[i-1]`
6. `p[i]`
7. `i++`
8. `i--`
9. `>=0`
10. `>0`

3

121. Дате су следеће декларације: `int p[200], i, n, k, x;`

А затим и део кода који треба да у низ `p` дужине `n` УБАЦИ (инсертује) елеменат `x` на позицију `k`, а затим ажурира нову дужину низа.

Имајући у виду дату иницијализацију петље, у предвиђена поља унеси одговарајуће елементе **преписивањем израза** из листе понуђених израза (подразумевати да су све потребне променљиве иницијализоване):

```
for(i=n; i _____; _____)
    _____ = _____;
    _____ = x;
n++;
```

1. `p[i+1]`
2. `p[i-1]`
3. `p[i]`
4. `p[k]`
5. `i++`
6. `i--`
7. `>=k`
8. `>k`

3

122. Дате су следеће декларације: `int p[200], i, n, k;`

А затим и део кода који треба да из низа `p` дужине `n`, сажимањем ИЗБАЦИ елеменат низа са позиције `k`, а затим ажурира нову дужину низа.

Имајући у виду дату иницијализацију петље, у предвиђена поља унеси одговарајуће елементе **преписивањем израза** из листе понуђених израза (подразумевати да су све потребне променљиве иницијализоване):

```
for(i=k; i _____; _____)
    _____ = _____;
n--;
```

1. `p[i+1]`
2. `p[i-1]`
3. `p[i]`
4. `p[k]`
5. `i++`
6. `i--`
7. `<n`
8. `<n-1`

2

123. У програмском језику Ц, дате су следеће декларације: `int A[50], i, n;`
Потребно је формирти вектор са следећим вредностима:

i=0	i=1	i=2	i=3	i=4	i=5	...	i=n-1
1	2	4	7	11	16	...	???

Допунити програмски код којим се формира овај вектор:

```
A[0]=1;  
  
for(i = _____; i _____; i++)  
    _____ = _____;
```

3

124. У програмском језику Ц, декларисане су и иницијализоване променљиве:

```
int x=40, y=50, z=60, *p1, *p2;
```

Одреди које ће вредности имати променљиве x, y и z после извршења следећег кода и упиши на одговарајућу линију:

```
p1 = &x;  
p2 = p1;  
y = (*p2)+20;  
z = *p2;
```

x = _____; y = _____; z = _____;

3

125. Дат је део кода написан на програмском језику Ц:

```
int a[7]={10,25,30,15,40,77,45}, *pa, x, y;  
pa=a+4;  
x=--(*pa)+5;  
y=*(--pa)+5;
```

Анализирати код и одредити вредности променљивих x и y, као и показивача pa, по извршењу све три извршне наредбе датог кода:

x = _____

y = _____

pa = a + _____

3

126. Дат је део кода написан на програмском језику Ц:

```
int a[7]={81,12,35,97,40,52,17}, *pa, x, y;  
pa=a+3;  
x = *(pa-2)+1;  
y = (*pa-2)+1;
```

Анализирати код и одредити вредности променљивих x и y, као и показивача pa, по извршењу све три извршне наредбе датог кода:

x = _____

y = _____

pa = a + _____

3

127. Динамичку алокацију меморије за смештање једнодимензионалног низа елемената може се извршити следећим функцијама:

Функција _____ враћа показивач на додељени простор и при алокацији аутоматски врши попуњавање додељеног простора нулама.

1

Функција _____ враћа показивач на додељени простор, при чему додељени простор нема дефинисан садржај.

128. Наредбама програмског језика Ц декларисана је матрица и три целобројне променљиве:

```
int mat[50][50]; int k, N, Sum=0;
```

где променљива **N** представља димензију квадратне матрице **mat**.

Допунити изразима који недостају код петље која има задатак да одреди збир (**Sum**) свих елемената на споредној дијагонали матрице:

3

```
for (k=0; k< _____; k++)
    Sum = Sum + mat [ _____ ] [ _____ ];
```

129. Наредбама програмског језика Ц декларисана је правоугаона матрица и три целобројне променљиве:

```
int mat[10][20]; int x, N, M;
```

где **N** представља број врста, а **M** број колона правоугаоне матрице **mat**.

Допунити изразима који недостају код петље која има задатак да дуплира све елементе **последње колоне** матрице:

2

```
for (x = 0; x < _____; x++)
    mat [ _____ ] [ _____ ] *= 2;
```

130. Наредбама програмског језика Ц декларисана је правоугаона матрица и три целобројне променљиве:

```
int mat[10][20]; int k, N, M;
```

где **N** представља број врста, а **M** број колона правоугаоне матрице **mat**.

Допунити изразима који недостају код петље која има задатак да дуплира све елементе **прве врсте** матрице:

2

```
for (k=0; k< _____; k++)
    mat [ _____ ] [ _____ ] *= 2;
```

131. Дат је прототип функције написан синтаксом програмског језика Ц:

```
int Saberi(int a, int b, int c);
```

која сабира бројеве од **a** до **b** са кораком промене **c**. Написати позив функције којим се рачуна и целобројној променљивој **x** додељује вредност:

2

10+14+18+22+...+50

Позив функције:

132. Дата је дефиниција функције:

```
void Transformisi(float *x, float *y, float z)
{
    z++;
    *x = *x + z;
    (*y)++;
}
```

У главном програму су декларисане променљиве и извршен је позив функције:

```
float a=10, b=10, c=10;
Transformisi(&a, &b, c);
```

3

Одредити које вредности имају променљиве a, b и c по изласку из функције и уписати их на одговарајућу линију:

a = _____

b = _____

c = _____

133. Дата је дефиниција функције:

```
void Transformisi(float *a, float b)
{
    (*a)++;
    b=b+3;
}
```

У главном програму су декларисане променљиве и извршен је позив функције:

2

```
float x=4, y=7;
Transformisi(&x, y);
```

Одредити вредности променљивих x и y и уписати их на назначено место:

унутар функције, пре повратка у главни програм: x = _____, y = _____

по повратку из функције, у главном програму: x = _____, y = _____

134. На програмском језику Ц декларисани су и иницијализовани стрингови:

```
char s1[]{"crvena zvezda"};
char s2[]{"zelengora"};
```

Одредити и на предвиђену линију уписати садржај стрингова по извршењу следеће наредбе:

2

```
strncpy(s1,s2,3);
```

s1 = _____

s2 = _____

135. На програмском језику Ц декларисан је и иницијализован стринг и два показивача:

```
char s1[]{"Short Message Service", *s2, *s3};
```

Одредити и на предвиђену линију уписати садржај означених стрингова по извршењу следећих наредби:

```
s2=strchr(s1,'M');
s3=strrchr(s2,'S');
strncpy(s1+1,s2,1);
strcpy(s1+2,s3);
```

3

s1 = _____

s2 = _____

s3 = _____

136. На програмском језику Ц декларисани су стрингови:

```
char s1[100]="", s2[100]="biografija";
char *t="planarna geometrija";
```

Са леве стране написани су изрази. Одредити и на одговарајућу линију уписати тражену вредност стринга по извршењу наведене наредбе (наредбе не посматрати као секвенцу, већ независно једну од друге):

strcpy(s1, t);	s1= _____
strncpy(s1, t, 4);	s1= _____
strcpy(s2, t);	s2= _____
strncpy(s2, t+9, 3);	s2= _____

4

137. У програмском језику Ц дата је декларација структурног типа **Complex** који описује комплексни број, а затим декларисан показивач на бинарни фајл који у себи садржи комплексне бројеве:

```
typedef struct
{
    float Re, Im;
}Complex;...
```

```
FILE *bin;
```

Допунити програмски код којим, у отвореном бинарном фајлу врши позиционирање на последњи комплексни број запамћен у фајлу у циљу ажурирања његове вредности:

```
fseek(_____,_____,_____);
```

2

У следећим задацима уредите и повежите појмове према захтеву

138. Наведене су етапе у изради програма. Поређати етапе по редоследу њиховог изођења уписивањем редног броја на линију испред наведене етапе:

_____	кодирање
_____	израда документације
_____	анализа проблема
_____	превођење
_____	скицирање решења
_____	састављање алгоритма
_____	тестирање
_____	одржавање програма

3

139. Са леве стране су набројане етапе у изради програма, а са десне стране дати су типови грешака које се могу јавити приликом програмског решавања задатака. На линију испред назива грешке написати број којим је означена етапа у којој се дати тип грешака открива:

1. едитовање	_____	синтаксне грешке
2. лоадовање	_____	run-time грешке
3. тестирање	_____	логичке грешке
4. превођење	_____	

1,5

140.	Са леве стране су набројани неки од прелазних знакова тј. escape секвенце, а са десне стране дати су њихови описи. На линију испред описа упишите број под којим је наведена одговарајућа escape секвенца:		
1.	'\n'	враћање на почетак реда (carriage return)	3
2.	'\t'	системски звучник (bell)	
3.	'\r'	прелаз у нови ред (new line)	
4.	'\b'	није escape секвенца	
5.	'\h'	хоризонтални табулатор (horizontal tab)	
6.	'\a'	враћање једну курсорску позицију назад (backspace)	
141.	Декларисана је реална променљива <code>float w=123.456;</code> :		
Са леве стране су дати различити прикази вредности пременљиве добијени коришћењем наредби форматираног излаза које су приказане са десне стране. Поред сваке наредбе, на предвиђену линију уписати рени број приказа добијеног изршавањем те наредбе:			
1.	123.456000	<code>printf("%g", w);</code>	2
2.	1.234560e+002	<code>printf("%f", w);</code>	
3.	123.456	<code>printf("%.2f", w);</code>	
4.	123.46	<code>printf("%e", w);</code>	
142.	Са леве страни дате су врсте конверзије, а са десне типови података који се користе у функцији за приказ података printf у програмском језику Ц. На линију испред типа података унеси број којим је означена одговарајућа конверзија:		
1.	%d	short	
2.	%i	signed int (у декадном облику)	
3.	%s	long	2,5
4.	%ld	unsigned	
5.	%f	signed int (декадни, хексадекадни или октални облик)	
6.	%e		
7.	%hd		
8.	%u		
143.	Са десне стране дате су врсте конверзије, а са леве типови података који се користе у функцији за приказ података printf у програмском језику Ц. На линију испред типа конверзије унеси број којим је означен одговарајући тип података:		
1.	char	%f	
2.	string	%s	3
3.	float	%g	
4.	double	%lf	
5.	long	%e	
6.	short	%c	

144. Са леве стране дати су допунски параметри у функцији `printf`, а са десне стране значење тих параметара у програмском језику Ц. На линију испред значења унети број којим је означен одговарајући допунски параметар:

1. (#) _____ означава да ће се поравнавање вршити уз леву ивицу поља ширине **n** знакова, допунски знакови размака додају се иза, а не испред податка
2. (0) _____ означава да се испред позитивног броја мора исписати знак плус
3. (-) _____ нула код нумеричких података означава да ће се приликом равнања уз десну ивицу број допуњавати нулама, а не знаковима размак
4. (+) _____ исписује се децимална тачка у конверзији рационалних бројева који немају разломљени део.

2

145. Са леве стране дати су математички изрази, а са десне запис израза на програмском језику Ц. На линију испред записа израза унеси број којим је означен одговарајући израз:

1. $y = \frac{\sqrt{x+10}}{a + |b|}$ _____ $y = \text{sqrt}(x+10) / (\text{a}+\text{fabs}(b))$
2. $y = \frac{\sqrt{x+10}}{a} + |b|$ _____ $y = \text{sqrt}(x)+10 / \text{a}+\text{fabs}(b)$
3. $y = \frac{\sqrt{x+10}}{a + |b|}$ _____ $y = \text{sqrt}(x+10) / \text{a}+\text{fabs}(b)$
4. $y = \sqrt{x} + \frac{10}{a} + |b|$ _____ $y = (\text{sqrt}(x)+10) / (\text{a}+\text{fabs}(b))$

2

146. Дат је код на програмском језику Ц:

```
switch(c) {
    case 'A': case 'a': printf("Pravougaonik ");
    case 'B': case 'b': printf("Trougao "); break;
    case 'C': case 'c': printf("Krug ");
    default: printf("Duz "); break;
}
```

Са десне стране су дате вредности променљиве `c` (скретница), а са леве стране резултат извршења кода за дату вредност скретнице. На линију испред вредности скретнице унети редни под којим је наведен одговарајући екрански приказ:

2

1. Krug Duz _____ 'b'
2. Pravougaonik Trougao Krug Duz _____ 'K'
3. Krug _____ 'A'
4. Trougao _____ 'c'
5. Pravougaonik Trougao
6. Duz

147. Са леве стране су набројани различити типови променљивих, а са десне су дате декларације променљивих у програмском језику Ц. На линију испред декларације унети редни број под којим је наведен одговарајући тип променљиве:

1. Једнодимензионални низ показивача на целе бројеве _____ `int *a;`
2. Вектор целих бројева _____ `int a[100];`
3. Показивач на цео број _____ `int a*[100];`
4. Цео број _____ `int *a[100];`
5. Грешка у декларацији

2

148. Са леве стране су дате декларације променљивих у програмском језику Ц, а са десне наведена величина меморијског простора која се резервише при извршењу датих декларативних наредби. На линију испред декларације унети редни број под којим је наведен одговарајући капацитет меморијског простора:

<u>_____</u>	<code>int *a;</code>	1. 128 бајтова	2
<u>_____</u>	<code>char a[64];</code>	2. 256 бајтова	
<u>_____</u>	<code>float a[32];</code>	3. 32 бајта	
<u>_____</u>	<code>short *a;</code>	4. 64 бајта	
		5. 2 или 4 бајта, зависно од капацитета меморије	

149. У програмском језику Ц, декларисан је показивач на цео број и функцијом `calloc` додељен му је простор за смештај низа од `n` целих бројева:

```
int *a, n;
scanf("%d", &n);
a=(int*)calloc(n, sizeof(int));
```

У левој колони дати су изрази, а у десној опис њиховог значења. На линију испред сваког од израза унеси број којим је означен одговарајуће објашњење:

<u>_____</u>	<code>&a[0];</code>	1. вредност елемента на последњој позицији у низу	2
<u>_____</u>	<code>* (a+n-1);</code>	2. адреса четвртог елемента у низу	
<u>_____</u>	<code>a+4;</code>	3. адреса почетног елемента низа	
<u>_____</u>	<code>*a;</code>	4. вредност елемента на предзадњој позицији низа	
<u>_____</u>		5. вредност елемента на почетној позицији у низу	
<u>_____</u>		6. адреса петог елемента у низу	

150. Наредбама програмског језика Ц декларисано је дводиманзионално поље реалних бројева (матрица) и три целобројне променљиве:

```
float mat[10][10]; int i, j, n;
```

Где променљива `n` представља димензију квадратне матрице `mat`.

Са леве стране су дате ознаке елемената матрице, а са десне њихово тумачење. На линију испред сваке ознаке унети редни број одговарајућег тумачења:

<u>_____</u>	<code>mat[j][n-1]</code>	1. елеменат у j-тој врсти и последњој колони	2
<u>_____</u>	<code>mat[j]</code>	2. i-та врста матрице	
<u>_____</u>	<code>mat[0][j]</code>	3. j-та врста матрице	
<u>_____</u>	<code>mat[i]</code>	4. j-та колона матрице	
<u>_____</u>		5. елеменат у првој врсти и j-тој колони	
<u>_____</u>		6. грешка у нотацији	

151. Дат је део кода на програмском језику Ц. Одредити да ли се променљива **P** види и ако се види ког је типа у функцијама датог кода. Са леве стране набројане су функције из примера, а са десне тип променљиве **P** у поједним функцијама. На линију испред назива функције, уписати редни број под којим је дат одговарајући тип:

```
#include <stdio.h>
char P,Q;
_____  

float pp1(int n, int m)
{ float S=0, P=1; ... }  

_____  

int pp2(char *R, char *Q)
{ int i; ... }  

_____  

int pp3(float P[], int n)
{ int i; ... }  

_____  

main()
{ int S; ... }
```

У функцији pp1 1. карактер
У функцији pp2 2. реални број
У функцији pp3 3. низ реалних бројева
У функцији main 4. показивач на карактер
У функцији main 5. није доступна у функцији

2

4

3

152. На програмском језику Ц декларисане су променљиве:

```
char s1[]="Iwnt2CmyM8sagain", *sn;
```

Са леве стране написани су изрази доделе вредности стрингу **sn**, а са десне стране понуђене су вредности стринга **sn**. На линију написати редни број под којим је наведена вредност стринга **sn** која се добија извшењем одговарајућег израза:

_____	sn=strrchr(s3, 'a')-1;	1. NULL
_____	sn=strchr(s3, 'a')+1;	2. "in"
_____	sn=strstr(s3, "my");	3. "ain"
_____	sn=strstr(s3, "T2");	4. "gain"
		5. "sagain"
		6. "myM8sagain"

153. Са десне стране наведене су неке од функција библиотеке ctype.h, а са леве су дати њихови описи. Испред назива сваке од наведених функција, уписати редни број под којим је дат одговарајући опис:

- | | | | |
|----|---|-------|------------|
| 1. | Да ли је с штампајући знак (укључујући и размак)? | _____ | isspace(c) |
| 2. | Да ли је с велико слово? | _____ | isdigit(c) |
| 3. | Да ли је с знак интерпункције? | _____ | isalpha(c) |
| 4. | Да ли је с управљачки знак? | _____ | isupper(c) |
| 5. | Да ли је с децимална цифра? | _____ | iscntrl(c) |
| 6. | Да ли је с знак бели знак? | _____ | isprint(c) |
| 7. | Да ли је с слово? | | |
| 8. | Да ли је с хекса-децимална цифра? | | |

154.	Са леве стране наведене су ознаке за мод приступа датотеци, а са десне стране значење мода приступа. На линију испред значења мода приступа унети редни број под којим је наведена ознака одговарајућег мода:	
1.	"a"	Отвара текстуалну датотеку само за читање Отвара текстуалну датотеку само за писање. Ако датотека постоји одбације постојећи садржај без опомене, ако не постоји креира нову датотеку
2.	"w"	Отвара текстуалну датотеку само за писање додајући нове елементе на њен крај или креира потпуно нову датотеку ако је нема
3.	"a+"	Отвара текстуалну датотеку за ажурирање, читање и писање постављајући показивач на почетак датотеке
4.	"r+"	Отвара текстуалну датотеку за ажурирање, читање и писање, одбацијући постојећи садржај ако датотека постоји или креира датотеку ако она не постоји
5.	"r"	Отвара текстуалну датотеку за ажурирање, читање и писање додајући нове елементе на њен крај или ствара нову датотеку ако је нема
6.	"w+"	
155.	Са леве стране наведене су функције за читање и упис у текст датотеку, а са десне стране опис функције. На линију испред описа функције унети редни број под којим је наведена одговарајућа функција:	
1.	fscanf	учитавање карактера из датотеке
2.	fgets	учитавање реда из датотеке
3.	fputs	форматирани упис података у датотеку
4.	fprintf	упис стринга у датотеку
5.	fgetc	форматирано учитавање података из датотеке
156.	Наведене су карактеристике бинарних и текстуалних датотека. На линију испред сваке од наведених карактеристика уписати број 1 ако се карактеристика односи на бинарне, односно 2 ако се односи на текстуалне датотеке:	
1.	Бинарна	садржај датотеке може командама оперативног система да буде приказан на екрану или одштампан на штампачу
2.	Текстуална	приликом преноса се не примењује конверзија, већ се подаци преносе бајт по бајт
3.		организоване су у линије
4.		не могу се без конверзије послати на излазне уређаје, нити се преузети са улазних уређаја
157.	Са леве стране су наведене наредбе позиционирања у датотеци, а са десне описи ефекта датих наредби. На линију поред наредбе уписати редни број под којим је наведен опис ефекта наредбе:	
1.	ftell(dat)	позиционирање на почетак датотеке
2.	fseek(dat, 0, SEEK_END)	позиционирање на крај датотеке
3.	fseek(dat, 0, SEEK_SET)	одређује позицију у датотеци у виду броја бајтова од почетка датотеке
4.	rewind(dat)	ништа од понуђеног

ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК С#

У следећим задацима заокружите број испред траженог одговора

158. Дати су типови променљивих у програмском језику С#. Одредити како се назива променљива која је дефинисана унутар неког метода.
Заокружити број испред очекиваног одговора:
1. Глобална променљива
 2. Статичка променљива
 3. Блоковска променљива
 4. Локална променљива
- 1
159. Дате су програмске јединице у програмском језику С#. Одредити која програмска јединица представља шаблон за конструисање објекта истог типа. Заокружити број испред очекиваног одговора:
1. Класа
 2. Именски простор
 3. Нестатичка метода
 4. Променљива
 5. Статичка метода
- 1
160. Одредити какви могу бити чланови класе (поља и методе) у програмском језику С#. Заокружити број испред очекиваног одговора:
1. Локални и глобални
 2. Процедурални и непроцедурални
 3. Статички (класни) и нестатички (објектни)
 4. Спољашњи и унутрашњи
- 1
161. Дат је код програма у програмском језику С#. Анализирати дати код и проценити његову тачност. Заокружити број испред понуђеног тачног исказа:
- ```
namespace TestPrimer {
 class Test {
 int x;
 public Test(string s){
 Console.WriteLine("Klasa Test");
 }
 static void Main(string[] args){
 Test t = null;
 Console.WriteLine(t.x);
 }
 }
}
```
- 2
1. Програм има грешку јер променљива x није иницијализована.
  2. Програм има грешку јер класа Test нема подразумевани конструктор.
  3. Програм има грешку јер се у некој класи не може декларисати променљива типа те исте класе, као што је то овде случај са променљивом t.
  4. Програм има грешку јер променљива t није иницијализована и има вредност **null** у моменту када се приказује поље **t.x**.
  5. Програм нема грешака и нормално се извршава, не приказујући ништа на екрану.

162. Дати код програма у програмском језику C# састоји се од две класе у једној датотеци. Анализирати дати код и проценити његову тачност.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 Klasa a = new Klasa();
 a.n++;
 }
 }
 class Klasa {
 int n;
 private Klasa() { }
 }
}
```

Заокружити број испред тачног исказа:

1. Програм има грешку јер класа **Klasa** има приватни конструктор и приватно поље **n**.
2. Програм има грешку јер класа **Klasa** има празан подразумевани конструктор.
3. Програм има грешку јер променљива **n** није иницијализована.
4. Програм нема грешака и нормално се извршава

2

163. Дат је код програма у програмском језику C#:

```
namespace TestPrimer {
 class Program
 {
 static void Main(string[] args)
 {
 Console.WriteLine(fun(17));
 }
 public int fun(int n) { return n; }
 public void fun(int n){ Console.WriteLine(n); }
 }
}
```

2

Анализирати код и заокружити број испред очекиваног одговора:

1. Програм има грешку, јер се не може одредити коју верзију преоптерећеног метода **fun(...)** треба позвати.
2. Програм има грешку, јер је друга верзија преоптерећеног метода **fun(...)** дефинисана али се нигде не позива.
3. Програм се нормално извршава и приказује 17 једанпут.
4. Програм се нормално извршава и приказује 17 двапут.

164. У програмском језику C# дана је декларација низа:

```
int k;
int[] brojevi = {5, 12, 37, 7, 27, 33, 36};
```

На основу дате декларације одредити шта је резултат позива

**k=Arrays.BinarySearch(brojevi, 37);**

Заокружити број испред очекиваног одговора:

1. k=0, јер метод BinarySearch прво изврши сортирање низа у опадајућем редоследу, па онда тражи задату вредност
2. метод BinarySearch баца изузетак увек када је низ неуређен и програм „пуца“
3. k=2, јер се тражени елеменат налази на позицији 2
4. k добија неочекивану вредност јер низ мора бити сортиран у растућем поретку пре позива методе BinarySearch
5. k=6, јер метод BinarySearch прво изврши сортирање низа у растућем редоследу, па онда тражи задату вредност

2

165. У програмском језику C# декларисане су статичке променљиве логичког, нумеричког и класног типа.

```
namespace Test {
 class Program
 {
 public static bool x;
 public static int y;
 public static Random r;
 static void Main(string[] args) { ... }
 }
}
```

Одредити које аутоматске почетне (default) вредности декларисана поља имају унутар методе Main. Понуђени одговори су наведени у наглашеном редоследу. Заокружити број испред очекиваног одговора:

|    | x     | y | r    |
|----|-------|---|------|
| 1. | true  | 1 | null |
| 2. | false | 0 | null |
| 3. | true  | 0 | null |
| 4. | false | 1 | null |
| 5. | false | 0 | void |

1

166. Дат је код програма у програмском језику C# и састоји се од две класе у једној датотеци. Анализирати дати код и проценити која се вредност поља b.n приказује првом наредбом `Console.WriteLine` (ред седам) приликом извршавања овог програма. Заокружити број испред очекиваног одговора:

```
1. namespace TestPrimer {
2. class Program {
3. static void Main(string[] args){
4. int k = 0;
5. Brojac b = new Brojac();
6. for (int i = 0; i < 100; i++) Inc(b, k);
7. Console.WriteLine("b.n = " + b.n);
8. Console.WriteLine("k = " + k);
9. }
10. public static void Inc(Brojac b, int k){
11. b.n++;
12. k++;
13. }
14. }
15. class Brojac {
16. public int n;
17. public Brojac(int n) { this.n = n; }
18. public Brojac() { this.n = 1; }
19. }
20. }
```

2

1. b.n = 101
2. b.n = 100
3. b.n = 99
4. b.n = 0
5. b.n = 1

167. У програмском језику C# дефинисана је метода са аргументом низовног типа. Одредити шта се тачно преноси том методу. Заокружити број испред очекиваног одговора:

1. Копија датог низа.
2. Копија првог елемента датог низа.
3. Референца на дати низ.
4. Дужина датог низа

1

168. Дата је наредба кода у програмском језику C# која представља дефиницију низа **niz**. Проценити која од доле наведених дефиниција низа је коректна. Заокружити број испред очекиваног одговора:

1. int[] niz = new int[20];
2. int[] niz = new int(20);
3. int niz = new int[20];
4. int niz() = new int[20];

1

169. Дат је код програма у програмском језику C# који формира и штампа елементе низа **a**. Анализирати дати код и проценити шта ће се догодити након његовог извршавања.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 int[] a = new int[5];
 for (int i = 0; i < a.Length; i++) a[i] = i;
 Console.WriteLine(a[i] + " ");
 }
 }
}
```

2

Заокружити број испред очекиваног одговора:

1. Програм приказује бројеве 0 1 2 3 4 на екрану.
2. Програм има грешку, јер ће у последњој наредби **Console.WriteLine** метода **Main** покушати приступ непостојећем елементу **a[5]**.
3. Програм приказује број 5 на екрану.
4. Програм има грешку, јер променљива **i** у последњој наредби **Console.WriteLine** у методу **Main** неће имати дефинисану вредност.

170. Дат је код програма у програмском језику C#. Анализирати програм и одредити шта се приказује на екрану као резултат његовог извршавања.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 int[] x = { 0, 1, 2, 3, 4 };
 Inc(x);
 int[] y = { 0, 1, 2, 3, 4 };
 Inc(y[0]);
 Console.WriteLine(x[0] + " " + y[0]);
 }
 public static void Inc(int[] a) {
 for (int i = 0; i < a.Length; i++) a[i]++;
 }
 public static void Inc(int n) { n++; }
 }
}
```

2

Заокружити број испред очекиваног одговора:

1. Порука о грешци
2. 1 0
3. 2 2
4. 2 1
5. 1 1

171. Заокружити број испред исправно написане наредбе кода у програмском језику C# којом се копија низа x додељује низу y:

1. `Array.Copy(x, y, x.Length);`
2. `Array.Copy(x, y);`
3. `y = Array.Copy(x, x.Length);`
4. `y = x.CopyTo();`

1

172. Дат је програм у програмском језику C# у ком је дефинисан низ и метода **Prosek** која израчунава средњу вредност низа. Анализирати дати код и проценити шта ће се десити након његовог извршавања.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args){
 double[] niz = { 1.0, 2.0, 3.0 };
 Console.WriteLine(prosek(niz));
 Console.WriteLine(prosek(1, 2, 2, 1, 4));
 Console.WriteLine(prosek(new double[] { 1, 2, 3 }));
 Console.WriteLine(prosek(1.0, 2.0, 2.0, 1.0));
 }
 public static double prospek(params double[] brojevi){
 double suma = 0;
 foreach (double elem in brojevi) suma += elem;
 return suma / brojevi.Length;
 }
 }
}
```

2

Заокружити број испред очекиваног одговора:

1. Програм има грешку у првој наредби WriteLine, јер је неправилан позив функције **prosek(niz)**.
2. Програм има грешку у другој наредби WriteLine, јер је неправилан позив функције **prosek(1, 2, 2, 1, 4)**.
3. Програм има грешку у трећој наредби WriteLine, јер је неправилан позив функције **prosek(new double[] {1, 2, 3})**.
4. Програм има грешку у четвртој наредби WriteLine, јер је неправилан позив функције **prosek(1.0, 2.0, 2.0, 1.0)**.
5. Програм се извршава без грешке и просек датих бројева се тачно израчунава.
6. Програм се извршава без грешке, али се просек датих бројева не израчунава тачно.

173. У програмском језику C# дата је декларација једне стринг и једне целебројне променљиве, као и део кода:

```
string str = "Primer";
int broj = 66;
Console.WriteLine(str + broj + 65);
Console.WriteLine(broj + 65 + str);
```

2

Анализирати код и проценити шта ће се приказати на екрану након његовог извршења.  
Заокружити број испред очекиваног одговора:

- |                            |                             |                           |                         |
|----------------------------|-----------------------------|---------------------------|-------------------------|
| 1. Primer6665<br>131Primer | 2. Primer6665<br>6665Primer | 3. Primer131<br>131Primer | 4. PrimerBA<br>BAPrimer |
|----------------------------|-----------------------------|---------------------------|-------------------------|

174. Дат је део кода који је написан на C# програмском језику. Анализирати код и одредити шта ће бити на излазу.

```
class Test
{
 public static void Main(string[] args) {
 Test p = new Test();
 p.start();
 }
 void start() {
 bool b1 = false;
 bool b2 = fix(b1);
 Console.WriteLine(b1 + " " + b2);
 }
 bool fix(bool b1) {
 b1 = true;
 return b1;
 }
}
```

2

Заокружи број испред тачног одговора:

1. true true
2. false true
3. true false
4. false false

175. Дата је наредба кода у програмском језику C#, која представља позив методе класе Array. Одредити којим од датих позива се сортира низ **brojevi** типа int[]. Заокружити број испред очекиваног одговора:

1. Array(brojevi)
2. brojevi.Sort()
3. Array.SortArray(brojevi)
4. Array.Sort(brojevi)

1

176. На програмском језику C# дефинисане су класе:

```
public class A {
 public virtual int Metod() { return 10; }
}
public class B : A {
 public new int Metod() { return base.Metod() + 20; }
}
public class C : B {
 public override int Metod() { return base.Metod() + 30; }
}
```

2

Унутар функције Main, креирана су три објекта ових класа и из сваког од њих извршен позив методе **Metod()**

Заокружити број испред понуђеног одговора који по вашем мишљењу представља сценарио који ће се десити при покретању програма:

1. Метод позван из класе А враћа вредност 10, из класе В 10, а из класе С 30
2. Метод позван из класе А враћа вредност 10, из класе В 30, а из класе С 60
3. Програм се покреће, али баца изузетак при позиву методе из класе В јер new не иде у комбинацији са virtual
4. Компилатор јавља грешку јер у класи С метод не може бити редефинисан

177. У програмском језику C#, по дефиницији метод може да врати вредност, али и не мора. Изабрати службену реч која се користи за означавање типа резултата методе када дефинишемо метод који НЕ враћа вредност. Заокружити број испред очекиваног одговора.

1. void
2. return
3. public
4. static

1

178. У програмском језику C# дат је метод **PrintPoruka** који је дефинисан на следећи начин:

```
void PrintPoruka(string poruka, int n) {
 while (n > 0) {
 Console.WriteLine(poruka);
 n--;
 }
}
```

Одредити шта ће бити исписано на екрану након позива метода **PrintPoruka("b",3)**. Заокружити број испред очекиваног одговора:

1

1. На екрану се приказује bbbbb
2. На екрану се приказује bbbb
3. На екрану се приказује bbb
4. На екрану се приказује bb
5. На екрану се приказује b

179. У програмском језику C# дат је метод **PrintPoruka** који је дефинисан на следећи начин:

```
void PrintPoruka(string poruka, int n) {
 while (n > 0) {
 n--;
 Console.WriteLine(poruka);
 }
}
```

Израчунати вредност променљиве m након извршења следећег програмског фрагмената:

1

```
int m = 2;
PrintPoruka("C# !!!", m);
```

Заокружити број испред очекиваног одговора:

1. m =0
2. m =1
3. m =2
4. m =3

180. Дат је код у програмском језику C#, који дефинише рекурзивни метод. Анализирати код и одредити резултат извршавања задатог метода.

```
public long fun(int n) {
 return n * fun(n - 1);
}
```

Заокружити број испред очекиваног одговора:

2

1. Резултат позива fun(3) је 1.
2. Резултат позива fun(3) је 2.
3. Резултат позива fun(3) је 6.
4. Позив fun(3) изазива грешку јер производи бесконачан ланац позива истог метода fun(...).

181. Дат је код у програмском језику C#, који дефинише рекурзивни метод. Анализирати код и одредити резултат који ће се приказати на екрану.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 fun(2);
 }

 public static void fun(int n) {
 while (n > 1) {
 Console.Write((n - 1) + " ");
 fun(n - 1);
 }
 }
 }
}
```

Заокружити број испред очекиваног одговора:

2

1. Програм на екрану не приказује ништа
2. Програм на екрану приказује 1 2 3
3. Програм на екрану приказује 3 2 1.
4. Програм на екрану бесконачно приказује 1 1 1 1 1 ....
5. Програм на екрану бесконачно приказује 2 2 2 2 2 ....

182. Дат је код у програмском језику C# који дефинише рекурзивни метод. Анализирати код и одредити резултат извршавања задатог метода.

Заокружити број испред очекиваног одговора:

2

```
public static int fun(int n) {
 if (n == 1) return 1;
 else return n + fun(n - 1);
}
```

1. Позивом fun (4) се исти метод fun () позива још 2 пута.
2. Позивом fun (5) се исти метод fun () позива још 4 пута.
3. Позивом fun (4) се исти метод fun () позива још 4 пута.
4. Позивом fun (5) се исти метод fun () позива још 6 пута.

183. У програмском језику C# дат је рекурзивни метод који проверава да ли је неки стринг палиндром. Да би код био комплетиран потребно је допунити трећи ред условом **if** наредбе.

```
1. public static bool palindrom(String s)
2. {
3. if (s.Length <= 1) return true; //bazni slučaj
4. else if (_____) return false;
5. else return palindrom(s.Substring(1, s.Length - 2));
6. }
```

2

Заокружити број испред очекиваног одговора:

1. s[0] != s[s.Length - 1]
2. s[0] != s[s.Length]
3. s[1] != s[s.Length - 1]
4. s[1] != s[s.Length]

184. Заокружити број испред појма чији је опис дат у тексту питања.

Процес скривања неких података о објекту од корисника и пружања само неопходних података, зове се...

1. Полиморфизам
2. Наслеђивање
3. Енкапсулација
4. Апстракција

1

185. У програмском језику С# дат је рекурзивни метод који проверава да ли је неки стринг палиндром. Да би код био комплетиран потребно је допунити седми ред.

```
1. public static bool Palindrom(String s){
2. return Palindrom(s, 0, s.Length - 1);
3. }
4. public static bool Palindrom(String s, int levi, int desni){
5. if (desni <= levi) return true; // bazni slucaj
6. else if (s[levi] != s[desni]) return false;
7. else return _____;
8. }
```

2

Заокружити број испред очекиваног одговора:

1. Palindrom(s)
2. Palindrom(s, levi, desni)
3. Palindrom(s, levi + 1, desni - 1)
4. Palindrom(s, levi + 1, desni)
5. Palindrom(s, levi, desni - 1)

186. У програмском језику С# дат је рекурзивни метод за сортирање низа рационалних бројева у опадајућем редоследу. Да би код био комплетиран потребно је допунити други ред помоћу једног од понуђених одговора.

```
1. public static void Sortiranje(double[] niz){
2. _____;
3. }
4. public static void Sortiranje(double[] niz, int kraj){
5. if (kraj > 0){
6. int imin = 0;
7. double min = niz[0];
8. for (int i = 1; i <= kraj; i++)
9. if (niz[i] < min){
10. min = niz[i];
11. imin = i;
12. }
13. niz[imin] = niz[kraj];
14. niz[kraj] = min;
15. Sortiranje(niz, kraj - 1);
16. }
17. }
```

2

Заокружити број испред очекиваног одговора:

1. Sortiranje(niz)
2. Sortiranje(niz, niz.length)
3. Sortiranje(niz, niz.length+1)
4. Sortiranje(niz, niz.length-1)

187. У програмском језику С# дат је рекурзивни метод за бинарно претраживање сортираног целобројног низа. Да би код био комплетиран потребно је допунити девети ред (означен линијом) помоћу понуђеног одговора.

2

```

1. public static int TraziBroj(int[] niz, int broj) {
2. return TraziBroj(niz, broj, 0, niz.Length - 1);
3. }
4. public static int TraziBroj(int[] niz,int broj,int levi,int desni) {
5. if(levi > desni) return -1; // broj nije nadjen u nizu
6. int sredina = (levi + desni) / 2;
7. if(broj < niz[sredina]) return TraziBroj(niz, broj, levi,sredina-1);
8. else if(broj > niz[sredina]) return _____;
9. else return sredina;
10. }

```

Заокружити број испред очекиваног одговора:

1. TraziBroj(niz, broj, sredina + 1, levi)
2. TraziBroj(niz, broj, sredina - 1, levi)
3. TraziBroj(niz, broj, desni, sredina + 1)
4. TraziBroj(niz, broj, sredina + 1, desni)

188. Одредити која поља су заједничка и јединствена за све креиране објекте неке класе дефинисане у објектно оријентисаном програмском језику C#.

Заокружити број испред очекиваног одговора:

1. Јавна
2. Приватна
3. Објектна
4. Инстанцна
5. Статичка

1

189. У програмском језику C# класа може да садржи статичка и не-статичка (инстанцна) поља. Дате су изјаве које се односе на статичка поља класе и међу њих је уметнута једна изјава која се односи на не-статичка (инстанцна) поља класе.

Заокружити број испред изјаве која се односи на не-статичка поља класе:

1. Поље које се може користити без конструисања иједног објекта те класе
2. Поље које има исту вредност за све креиране објекте неке класе
3. Поље чија вредност може разликовати за сваки појединачни објекат неке класе
4. Поље које се може користити унутар статичких метода класе, као и унутар метода инстанце

1

190. Дата је дефиниција класе у програмском језику C#. Проценити где у дефиницији класе (исpred које методе) треба заменити знакове ??? службеном речју **static**.

```

1. public class Test {
2. private int broj;
3.
4. public ??? int kvadrant(int n) { return n * n; }
5. public ??? int getBroj() { return broj; }
6. }

```

2

Заокружити број испред тачне изјаве:

1. Метода **kvadrant** МОРА да буде статичка, док метода **getBroj** може и не мора.
2. Обе методе морају бити статичке.
3. Ни једна од дефинисаних метода није статичка.
4. Метода **getBroj** НЕ СМЕ да буде статичка, док метода **kvadrant** може и не мора.

191. Дата је наредба кода у програмском језику C# која дефинише константу MAX\_CENA као чланицу неке класе. Проценити који од понуђених начина дефинисања је исправан.

```
1. public static const MAX_CENA = 100.00;
2. public const float MAX_CENA = 99.50;
3. public static double MAX_CENA = 100.50;
4. public static const double MAX_CENA = 100.99;
5. public const double MAX_CENA = 100.99;
```

1

192. Дат код програма у програмском језику C#. Анализирати дати код и проценити његову тачност. Заокружити број испред очекиваног одговора:

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 int n = 2;
 fun(n);
 Console.WriteLine("n je " + n);
 }
 void fun(int n) { n++; }
 }
}
```

2

- 1. Програм има грешку, јер метод fun() не враћа ниједну вредност.
  - 2. Програм има грешку, јер метод fun() није дефинисан да буде статички.
  - 3. Програм приказује 1 на екрану.
  - 4. Програм приказује 2 на екрану.
  - 5. Програм приказује 3 на екрану.

193. Дата је дефиниција класе у програмском језику C# и састоји се од два конструктора, методе и поља x и y. У шестом реду написати конструктор копије објекта класе Point.

```
1. public class Point {
2. private double x, y;
3. public Point() { x = 0; y = 0; }
4. public void Set(double xx, double yy) { x=xx; y=yy; }
5. public Point(Point p) {
6. _____ //Odgovor
7. }
8. }
```

2

Закружити број испред очекиваног одговора:

1. `this(p.x, p.y);`
  2. `this(p);`
  3. `Set(p);`
  4. `Set(p.x, p.y);`

194. У програмском језику C#, метод ***Equals(...)*** је метод инстанце класе ***object*** којим се проверава да ли је објекат из кога се метод позива једнак неком задатом објекту.

Овај метод се може надјачати (*override-овати*) у наслеђеним класама. Одредити заглавље овог метода у класи **string** у којој би метод био надјачан. Заокружити број испред понуђеног тачног одговора:

1. public override bool Equals(string s)
  2. public new bool Equals(string s)
  3. public override bool Equals(object obj)
  4. public static bool Equals(object obj)
  5. public bool Equals(string s1, string s2)

2

195. У програмском језику C#, метод ***Equals()*** за проверу једнакости два објекта је дефинисан у класи ***Object***. У датом програмском коду у класи ***Klasa*** је надјачан (override-ован) метод ***Equals()***. Анализирати код и проценити тачност извршења.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 Object obj1 = new Klasa();
 Object obj2 = new Klasa();
 Console.WriteLine(obj1.Equals(obj2));
 }
 }
 class Klasa {
 int x;
 public override bool Equals(object o) {
 Klasa a = (Klasa)o;
 return this.x == a.x;
 }
 }
}
```

Заокружити број испред очекиваног одговора:

2

1. Програм има грешку, јер се изразом `obj1.Equals(obj2)` проверава једнакост објекта `obj1` и `obj2` различитог типа од `Object`.
2. Програм има грешку, јер се једнакост објекта `obj1` и `obj2` типа ***Klasa*** проверава изразом `obj1 == obj`.
3. Програм се извршава без грешке и приказује се ***true*** на екрану.
4. Програм се извршава без грешке и приказује се ***false*** на екрану

196. У програмском језику C#, метод ***Equals()*** за проверу једнакости два објекта је дефинисан у класи ***Object***. У датом програмском коду, у класи ***Klasa*** је предефинисан метод ***Equals()***. Анализирати код и проценити тачност извршења.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 Object obj1 = new Klasa();
 Object obj2 = new Klasa();
 Console.WriteLine(obj1.Equals(obj2));
 }
 }
 class Klasa {
 int x;
 public new bool Equals(Klasa o) {return this.x == a.x; }
 }
}
```

2

Заокружити број испред очекиваног одговора:

1. Програм има грешку, јер се изразом `obj1.Equals(obj2)` проверава једнакост објекта `obj1` и `obj2` различитог типа од `Object`.
2. Програм има грешку, јер се једнакост објекта `obj1` и `obj2` типа ***Klasa*** проверава изразом `obj1 == obj`.
3. Програм се извршава без грешке и приказује се ***true*** на екрану.
4. Програм се извршава без грешке и приказује се ***false*** на екрану

197. Дат је код програма у програмском језику C#. У Main() методи декларисане су променљиве **s**, **obj** и **t**. Анализирати декларацију и одредити на који објекат указују променљиве **s**, **obj** и **t**.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args)
 {
 string s = "c#";
 object obj = s;
 string t = (string) obj;
 }
 }
}
```

2

Заокружити број испред очекиваног одговора:

- Када се вредност променљиве **s** додељује променљивој **obj** у наредби **object obj = s**, конструише се нови објекат.
- Када се конвертује тип променљиве **obj** и њена вредност додељује променљивој **t** у наредби **string t = (string) obj**, конструише се нови објекат.
- Када се конвертује тип променљиве **obj** и њена вредност додељује променљивој **t** у наредби **string t = (string) obj**, садржај променљиве **obj** се мења.
- Променљиве **s**, **obj** и **t** указују на исти објекат типа **string**.

198. Дат је код програма у програмском језику C# који дефинише класу **Osoba** са њеним методама и атрибутима. Анализирати дати код и одредити који од понуђених одговора је потребно дописати у 1. ред кода како би метода била тачно дефинисана.

```
1. public class Osoba _____ {
2. private string ime;
3. private string prezime;
4. int godina;
5. public Osoba(string ime, string prezime, int godina) {
6. this.ime = ime;
7. this.prezime = prezime;
8. this.godina = godina;
9. }
10. public int GetGodina() { return this.godina; }
11. public int CompareTo(Osoba obj) {
12. if (this.godina > obj.GetGodina()) return 1;
13. else if (this.godina < obj.GetGodina()) return -1;
14. else return 0;
15. }
16. public override string ToString() {
17. return "Ime: " + this.ime + "\tPrezime: " +
18. this.prezime + "\tGodina: " + this.godina;
19. }
}
```

1

Заокружити број испред тачног одговора:

- : IComparable<Osoba>
- : Comparable
- : IEquatable<Osoba>
- : Comparer

199. Дат је код програма у програмском језику C# који дефинише класу Osoba са њеним методама и атрибутима. Анализирати дати код и на основу декларације метода CompareTo(...) и Clone() одредити код које недостаје у првој линији.

```

1. class Osoba : _____ {
2. private string ime;
3. private double dohodak;
4. public Osoba(string ime, double dohodak) {
5. this.ime = ime;
6. this.dohodak = dohodak;
7. }
8. public int CompareTo(Osoba osb) {
9. if (this.dohodak < osb.dohodak) return -1;
10. else if (this.dohodak > osb.dohodak) return 1;
11. else return 0;
12. }
13. public Object Clone() {
14. return this.MemberwiseClone();
15. }
16. public override string ToString() {
17. return "Ime: "+this.ime+"\nDohodak: "+this.dohodak;
18. }
19. }
```

2

Заокружити број испред тачног одговора:

1. IComparable<Osoba>, ICloneable
2. Comparable<Osoba>, Cloneable
3. IComparable<Osoba>, ICloneable<Osoba>
4. IComparable, ICloneable

200. Дат је код програма у програмском језику C# којим су дефинисане две класе: `class Program` која садржи `Main(string[] args)` методу и `class A`. Анализирати дати код и одредити да ли је код исправно написан. Понуђени одговори дају опис последица извршавања овог кода. Заокружити број испред тачног исказа.

```

class Program {
 public static void Main(string[] args) {
 A a1 = new A();
 A a2 = new A();
 Console.WriteLine(a1.Equals(a2));
 }
}
class A {
 int x;
 public bool Equals(A a) {
 return this.x == a.x;
 }
}
```

2

1. Програм има грешку, јер се изразом `a1.Equals(a2)` проверава једнакост објеката `a1` и `a2` различитог типа од `Object`.
2. Програм има грешку, јер се једнакост објеката `a1` и `a2` типа `A` проверава изразом `a1 == a2`.
3. Програм се извршава без грешке и приказује се `true` на екрану.
4. Програм се извршава без грешке и приказује се `false` на екрану.

201. Дат је код програма у програмском језику C# којим су дефинисане три класе: `class Program` која садржи `Main(string[] args)` методу, `class Vozilo` и `class Auto`. Компајлер приликом превођења овог кода јавља грешку. Заокружити број испред одговора у коме је написано који редови су изазвали грешку приликом компајлирања.

```

1. class Program {
2.
3. public static void Main(string[] args) {
4. Vozilo v = new Vozilo(8);
5. v.vazi();
6. v.brojVrata = 2;
7. Vozilo bmw = new Auto(2, 4);
8. bmw.vazi();
9. Auto audi = new Auto(4);
10. Auto.vazi();
11. }
12. }
13. class Vozilo {
14. public int brojTockova;
15. public Vozilo(int t) {
16. brojTockova = t;
17. }
18. public void vazi() {
19. Console.WriteLine("Vožnja vozila");
20. }
21. }
22. class Auto : Vozilo {
23. public int brojVrata;
24. public Auto(int v, int t) : base(t)
25. {
26. brojVrata = v;
27. }
28. public void vazi() {
29. Console.WriteLine("Vožnja auta");
30. }
31. }
```

1. Грешке су у редовима 7, 10 и 16.
2. Грешке су у редовима 7, 9 и 22.
3. Грешке су у редовима 6, 7 и 10.
4. Грешке су у редовима 6, 9 и 10.
5. Грешке су у редовима 6, 9 и 24.
6. Грешке су у редовима 9, 10 и 24.

3

202. Дат је код програма у програмском језику C# који дефинише интерфејс **Poredjenje**. Интерфејс **Poredjenje** садржи декларацију методе `porediPovrsine()`, која пореди објекте типа `Figura`. У датом коду дописати у 3. линији кода наредбу која недостаје да би метода била исправно декларисана.

```

1. namespace figure
2.
3. public _____ {
4. int porediPovrsine(Figura fig);
5. }
```

1

Заокружити број испред одговора који даје исправно решење:

1. extends Poredjenje
2. interface Poredjenje
3. implements Poredjenje
4. abstract Poredjenje

203. Дат је код програма у програмском језику C# којим су дефинисане две класе: `class Program` која садржи `Main(string[] args)` методу и `class A`. Анализирати дати код и одредити да ли је код исправно написан. Понуђени одговори дају опис последица извршавања овог кода. Заокружити број испред тачног исказа.

```
class Program {
 public static void Main(string[] args) {
 Object a1 = new A();
 Object a2 = new A();
 Console.WriteLine(a1.Equals(a2));
 }
}
class A {
 int x;
 public bool Equals(A a) {
 return this.x == a.x;
 }
}
```

2

1. Програм има грешку, јер се изразом `a1.Equals(a2)` проверава једнакост објекта `a1` и `a2` различитог типа од `Object`.
2. Програм има грешку, јер се једнакост објекта `a1` и `a2` типа `A` проверава изразом `a1 == a2`.
3. Програм се извршава без грешке и приказује се `true` на екрану.
4. Програм се извршава без грешке и приказује се `false` на екрану.

204. Једна од основних особина објектно оријентисаног језика је наслеђивање. Дате су насловне линије дефиниције класе. Заокружити број испред дефиниције класе која се НЕ може наследити.

1. `class A {}`
2. `class A { private A(){ } }`
3. `sealed class A {}`
4. `class A { protected A(){ } }`

1

205. Дат је део кода који је написан на C# програмском језику. Анализирати и одредити шта ће се приказати на излазу извршавањем овог кода.

```
class PassA {
 public static void Main(string[] args) {
 PassA p = new PassA();
 p.start();
 }
 public void start() {
 long[] a1 = {3,4,5};
 long[] a2 = fix(a1);
 Console.Write(a1[0] + a1[1] + a1[2] + " ");
 Console.WriteLine(a2[0] + a2[1] + a2[2]);
 }
 long[] fix(long[] a3) {
 a3[1] = 7;
 return a3;
 }
}
```

3

Заокружи број испред тачног одговора:

1. 12 15
2. 15 15
3. 3 4 5 3 7 5
4. 3 7 5 3 7 5

206. Дат је део кода који је написан на C# програмском језику. Анализирати код и одредити шта ће бити на излазу.

```
class PassS {
 public static void Main(String[] args) {
 PassS p = new PassS();
 p.start();
 }
 void start() {
 String s1 = "slip";
 String s2 = fix(s1);
 Console.WriteLine(s1 + " " + s2);
 }
 String fix(String s1) {
 s1 = s1 + "stream";
 Console.WriteLine(s1 + " ");
 return "stream";
 }
}
```

2

Заокружки број испред тачног одговора:

1. slip stream
2. slipstream stream
3. stream slip stream
4. slipstream slip stream

207. Дате су наредбе које су написане на C# програмском језику. Анализирати дате наредбе и одредити која од њих је исправно написана. Заокружити број испред тачног одговора:

1. int a = Math.Abs(-5);
2. int b = Math.Abs(5.0);
3. int c = Math.Abs(5.5f);
4. int d = Math.Abs(5L);

1

208. Дати су позиви методе Math.Max дефинисане у C# програмском језику. Анализирати дате позиве и одредити који од њих су исправно написани. Заокружити број испред **НЕИСПРАВНО** написаног позива методе:

1. Math.Max(1, 4);
2. Math.Max(2.3, 5);
3. Math.Max(1, 3, 5, 7);
4. Math.Max(-1.5, -2.8f);

1

209. Заокружити број испред исказа који представља исправан наставак дате тврђње:

При креирању објекта изведене класе...

1. извршава се само конструктор изведене класе
2. прво се извршава конструктор родитељске класе, али само ако је позван кључном речју **base**
3. обавезно се прво извршава конструктор изведене, а потом конструктор родитељске класе
4. обавезно се прво извршава конструктор родитељске, а потом конструктор изведене класе

1

210. Дат је код програма у програмском језику C# који дефинише апстрактну класу **Figura**. У класи Figura написати статички иницијални блок у ком се иницијализује вредност атрибута **brojFigura**. Заокружити број испред наредбе коју треба додати у 7. линију кода, којом се иницијализује вредност статичког атрибута **brojFigura**.

```

1. public abstract class Figura {
2. protected static int brojFigura;
3. protected string naziv;
4. public Figura(string naziv) {
5. this.naziv = naziv;
6. }
7. public abstract double povrsinaFigure();
8. public abstract double obimFigure();
9.
10. public static void UvecajBrojFigura() {
11. brojFigura++;
12. }
13. public override string ToString() {
14. return "Figura ";
15. }
16. }
```

2

Заокружити број испред одговора који даје исправно решење:

1. static Figura() { brojFigura = 0; }
2. brojFigura = 0;
3. static { brojFigura = 0; }
4. public Figura() { brojFigura = 0; }

211. Дат је код програма у програмском језику C# у ком су дефинисане три класе: **class Program** која садржи **Main(string[] args)** методу, **class A** и **class B**. Анализирати дати код и одредити шта ће се приказати на екрану као резултат извршавања овог програма. Заокружити број испред одговора који приказује резултат исписа.

```

class Program : A {
 public static void Main(string[] args) {
 Program p = new Program();
 }
}
class A : B {
 public A() { Console.WriteLine("Pozvan podrazumevani konstruktor
klase A"); }
}
class B {
 public B() { Console.WriteLine("Pozvan podrazumevani konstruktor
klase B"); }
}
```

2

1. Ништа.
2. Позван подразумевани конструктор класе A
3. Позван подразумевани конструктор класе B
4. Позван подразумевани конструктор класе A и у другом реду Позван подразумевани конструктор класе B
5. Позван подразумевани конструктор класе B и у другом реду Позван подразумевани конструктор класе A

212. У програмском језику C# користи се службена реч **base**. Проценити који од наредних исказа који дефинишу дату службену реч **НИЈЕ** тачан.

Заокружити број испред очекиваног одговора:

1. Службена реч **base** може послужити за позивање конструктора родитељске класе.
2. Службена реч **base** може послужити за позивање приватних метода родитељске класе којима се другачије не може приступити.
3. Службена реч **base** може послужити за позивање заклоњеног метода родитељске класе.
4. Службена реч **base** може послужити за позивање заклоњеног поља родитељске класе.

1

213. Дат је код програма у програмском језику C# у ком су дефинисане три класе: **class Program** која садржи **Main(string[] args)** методу, **class A** и **class B**. Анализирати дати код и одредити да ли је код исправно написан.

Заокружити број испред исказа који даје информацију о тачности кода.

```
class Program {
 public static void Main(string[] args) {
 B b = new B();
 b.Metod(5);
 Console.WriteLine("b.i је " + b.CitajI());
 }
}
class A {
 int i;
 public int CitajI(){return i;}
 public void Metod(int i) { this.i = i; }
}
class B : A {
 public void Metod(string s){
 Console.WriteLine(s);
 }
}
```

2

1. Програм има грешку, јер је метод **Metod(int i)** надјачан (предефинисан) са различитим потписом у класи B.
2. Програм има грешку, јер се **b.Метод(5)** не може позвати пошто је метод **Metod(int i)** заклоњен у класи B.
3. Програм има грешку због **b.i**, јер је поље **i** неприступачно из класе B.
4. Програм нема грешке, јер наслеђени метод класе A, **Metod(int i)** није надјачан у класи B, већ је дефинисан преоптерећен метод **Metod(string s)**.

214. Заокружити број испред исказа који представља исправан наставак дате реченице:

Ако try-catch наредба има више catch блокова у којима „хватамо“ изузетак основне **Exception** класе, заједно са изузецима других класа изведенних из класе **Exceptions...**

1. онда се изузетак основне Exception класе може „хватати“ у било ком catch блоку (редослед није битан, битно је да се наведу све могуће грешке)
2. онда се изузетак основне Exception класе мора „хватати“ у последњем catch блоку
3. онда се изузетак основне Exception класе мора „хватати“ у првом catch блоку
4. основна Exception класа се не комбинује у истој наредби са класама изведеним из ње јер их оснавна класа „маскира“

2

215. Дат је део кода који је написан на C# програмском језику. Анализирати код и одредити шта ће се приказати на излазу.

```

try
{
 int x = 0;
 int y = 5 / x;
}
catch (Exception e)
{
 Console.WriteLine("Exception");
}
catch (ArithmetiсException ae)
{
 Console.WriteLine(" Arithmetic Exception");
}
Console.WriteLine("finished");

```

2

Заокружи број испред тачног одговора:

1. Приказује се текст: finished
2. Приказује се текст: Exception
3. Ништа. Дешава се грешка приликом компајлирања
4. Приказује се текст: Arithmetic Exception

216. Дат је део кода који је написан на C# програмском језику. Анализирати код и одредити шта ће бити на излазу.

```

public class RTExcept
{
 public static void throwit() {
 Console.WriteLine("throwit ");
 throw new ApplicationException();
 }
 public static void Main(String[] args) {
 try {
 Console.Write("hello ");
 throwit();
 }
 catch (Exception re) {Console.WriteLine("caught ");}
 finally { Console.Write("finally ");}
 Console.WriteLine("after ");
 }
}

```

2

Заокружи број испред тачног одговора:

1. hello throwit caught
2. Грешка приликом компајлирања
3. hello throwit *RuntimeException* caught after
4. hello throwit caught finally after

## У следећим задацима заокружите бројеве испред тражених одговора

217. Дати су искази који у програмском језику C# дефинишу конструктор. Заокружити бројеве испред очекиваних одговора:
1. Подразумевани конструктор без параметара се увек аутоматски додаје класи.
  2. Подразумевани конструктор без параметара се класи аутоматски додаје уколико у њој није експлицитно дефинисан ниједан конструктор.
  3. У класи се мора експлицитно дефинисати бар један конструктор.
  4. Конструктори немају тип резултата, чак ни **void**.
218. Дати код програма у програмском језику C# састоји се од две класе у једној датотеци. Анализирати дати код и проценити његову тачност. Заокружити бројеве испред тачних исказа:
- ```

namespace TestPrimer {
    class Program {
        static void Main(string[] args) {
            A a = new A();
            a.Print();
        }
    }
    class A {
        string s;
        public A(string s) { this.s = s; }
        public void Print(){
            Console.WriteLine(s);
        }
    }
}

```
1. Програм има грешку, јер класа A није јавна класа.
 2. Програм има грешку, јер класа A нема подразумевани конструктор.
 3. Програм нема грешака и нормално се извршава ништа не приказујући на екрану.
 4. Програм има грешку која се може исправити уколико се у четвртом реду наредба **A a = new A();** замени наредбом **A a = new A("poruka");**
219. Дата је дефиниција класе у програмском језику C# и састоји се од два конструктора и поља x и y. У четвртом реду допунити подразумевани конструктор без параметара класе Point, који формира тачку у координатном почетку, позивајући конструктор са параметрима.
- ```

1. public class Point {
2. private double x, y;
3. public Point(double x, double y) {
4. this.x = x; this.y = y;
5. }
6. public Point() _____
7. }

```
- Заокружити бројеве испред тачних одговора:
1. **public Point(): base(0, 0) { }**
  2. **public Point(): this(0, 0) { }**
  3. **public Point() { Point(0,0); }**
  4. **public Point(): this(0.0, 0.0) { }**
  5. **public Point() { Point(0.0, 0.0); }**

220. Дата је дефиниција класе у програмском језику C# и састоји се од два конструктора, једне методе и поља x. У дефиницији се користи службена реч **this**. Анализирати дати код и проценити тачност следећих исказа. Заокружити бројеве испред тачних исказа:

```
class TestPrimer {
 public double x;
 public TestPrimer(double x) {
 this.fun();
 this.x = x;
 }
 public TestPrimer() {
 Console.WriteLine("Podrazumevani konstruktor");
 this(23);
 }
 public void Fun() {
 Console.WriteLine("Poziv metoda fun()");
 }
}
```

2

1. `this.Fun()` у конструктору `TestPrimer(double x)` може се поједноставити и заменити само са `Fun()`.
2. `this.x` у конструктору `TestPrimer(double x)` може се поједноставити и заменити само са `x`.
3. позив конструктора `this(23)` унутар другог конструктора `TestPrimer()` је прво шта се извршава и мора се писати одмах после декларације `public TestPrimer():this(23)`
4. `this(23)` у конструктору `Test()` мора се заменити са прецизнијим изразом `this(23.0)`.

221. Дати су искази који дефинишу иницијализацију почетних вредности променљивих у програмском језику C#. Заокружити бројеве испред **НЕТАЧНИХ** исказа:

1. Локалне променљиве метода добијају аутоматски почетне вредности.
2. Глобалне променљиве (поља) класе добијају аутоматски почетне вредности.
3. Променљива неког примитивног типа садржи вредност тог примитивног типа.
4. Променљива неког класног типа садржи меморијску адресу на којој се налази објекат тог класног типа.
5. Променљивој класног типа може се доделити цео број који представља важећу меморијску адресу.
6. При додели примитивне промељиве (**a**), другој примитивној променљивој (**b**), врши се копирање вредности промељиве **a** у променљиву **b**, чиме се добијају две независне копије исте вредности

2

222. Дата је дефиниција класе у програмском језику C# и састоји се од два конструктора, методе и поља x и y. У петом реду дефинисан је конструктор са параметрима који формира тачку са координатама x и y. Заокружити наредбе којима се може допунити дефиниција конструктора.

```
1. public class Point {
2. private double x, y;
3. public Point() { x = 0; y = 0; }
4. public void set(double xx, double yy) { x = xx; y = yy; }
5. public Point(double x, double y) { _____; }
6. }
```

2

Заокружити бројеве испред тачних одговора:

1. `this.x=x; this.y=y;`
2. `x=x; y=y;`
3. `set(x,y);`
4. `set(this.x,this.y);`
5. `x=this.x; y=this.y;`

223. Дата је наредба кода у програмском језику C# која представља декларацију низа. Проценити које од доле наведених декларација су тачне.  
Заокружити бројеве испред очекиваних одговора:

1. int niz = new int(30);
2. double[ ] niz = new double[30];
3. int[ ] niz = { 3, 4, 3, 2 };
4. char[ ] niz = new char[ ];
5. char[ ] niz = new char { 'a', 'b', 'c', 'd' };
6. char[ ] niz = new char[ ] { 'a', 'b' };

1,5

224. Дати су делови кода у програмском језику C# који треба да рачунају збир елемената матрице **a**, декларисане на следећи начин: `int[,] a = new int[10, 10]`. Анализирати дате кодове и проценити који од предлога је тачан.

Заокружити бројеве испред очекиваних одговора:

1. 

```
int sum = 0;
for (int i = 0; i < b.Length; i++)
 for (int j = 0; j < b[i].Length; j++)
 sum3 += b[i][j];
```
2. 

```
int sum = 0;
foreach (int x in a) sum1 += x;
```
3. 

```
int sum = 0;
for (int i = 0; i < a.GetLength(0); i++)
 for (int j=0; j<a.GetLength(1); j++)
 sum2 += a[i,j];
```
4. 

```
int sum = 0;
foreach (int[] vrsta in b)
 foreach (int el in vrsta)
 sum4 += el;
```

2

225. Дати искази дефинишу особине наслеђивања. Заокружити бројеве испред тачних исказа:

1. Једна класа у C#-у може директно бити наслеђена из више класа.
2. Наслеђена класа садржи додатна поља и методе у односу на своју родитељску класу.
3. „Класа А наслеђује класу Б” значи да је А поткласа класе Б.
4. Ако класа А наслеђује класу Б, тада објекти класе А садрже сва поља и све методе класе Б.
5. Ако класа А наслеђује класу Б, тада се за сваки објекат класе А подразумева да је истовремено и објекат класе Б.
6. У C#-у је могуће из једне у другу класу наследити само методе и својства, али не и поља класе.

1,5

226. Да би наслеђени метод могао да се **редефинише** и тиме измене његова функционалност у класама наследницама, у родитељској класи испред ознаке повратног типа метода наводи се нека од понуђених кључних речи. Заокружити бројеве испред кључних речи које омогућавају редефинисање дефинисаног метода кроз ланац наслеђивања:

1. new
2. virtual
3. sealed
4. override
5. abstract
6. base
7. довољно је да буде public или protected

1,5

227. На програмском језику C# дефинисане су класе:

```

1. public class A {
2. public virtual int Metod() { return 10; }
3. }
4. public class B : A {
5. public new int Metod() { return 20; }
6. }
7. public class C : B {
8. public override int Metod() { return 30; }
9. }
```

Компилатор јавља грешку при превођењу овог кода коју је могуће решити на више начина у зависности од очекиваног ефекта.

Заокружити бројеве испред понуђених решења која ће, по вашем мишљењу, отклонити грешку у коду:

1. У 5. линији кода метод у класи В прогласити за **abstract** уместо **new**
2. У 5. линији кода кључну реч **new** заменити са **override**
3. У 7. линији кода класу С наследити из класе А, уместо из класе В
4. У 8. линији кода, иза декларације методе у класи С, позвати основни метод :**base ()**
5. У 8. линији кода обрисати кључну реч **override** и заменити је са **sealed**

2

228. У класи **Figura** дат је подразумевани (default) конструктор и конструктор са 4 параметра:

```
public Figura() {...}
public Figura(string ime, string boja, int pozX, int pozY) {...}
```

Заокружити бројеве испред исправно написаних наредби креирања објеката класе Figura:

1. **Figura f = Figura("lovac", "beli", 7, 3);**
2. **Figura f = new Figura("beli", "lovac", 7, 3);**
3. **Figura f = new Figura();**
4. **Figura f = new Figura("lovac", 3, 7, "beli");**
5. **Figura f = new Figura("lovac", "beli", 3, 7);**
6. **Figura f = new Figura("lovac", "beli", 3);**

3

229. Заокружити бројеве испред наведених чланова класе који се ни под којим условима **НЕ** наслеђују са родитељске класе на изведену класу:

1. Readonly својства
2. Заштићени чланови класа
3. Својства (property)
4. Приватни чланови класа
5. Конструктор класе

2

230. Дати су искази који програмском језику C# дефинишу рекурзивне методе.

Заокружити бројеве испред **НЕТАЧНИХ** исказа:

1. Рекурзивни методи су они који позивају сами себе, било директно или индиректно.
2. Рекурзивни методи се позивају другачије од нерекурзивних метода.
3. Применом рекурзије у решавању проблема увек долази до побољшања перформанси програма
4. Рекурзивни методи решавају неки задатак својењем полазног проблема на сличан, простији проблем (или више њих).
5. Сваки рекурзивни метод мора имати базни случај за најпростији задатак чије се решење не добија рекурзивним позивом.

2

231. Дате су наредбе у програмском језику C# које дефинишу заглавље методе Print() са променљивим бројем параметара. Одредити који од понуђених одговора су исправни. Заокружити бројеве испред очекиваних одговора:

1. `public void Print(params string[] niska, params double[] broj)`
2. `public void Print(params double[] broj, string niska)`
3. `public void params Print(double d1, double d2)`
4. `public void Print(params double[] broj)`
5. `public void Print(int n, params double[] broj)`

2

232. Дати су искази који дефинишу полиморфизам, једну од особина објектно-оријентисаног језика C#. Анализирати исказе и заокружити бројеве испред тачних исказа:

1. Када је класа Б изведена из класе А, објекту класе А може се доделити показивач на објекат класе Б
2. Када је класа Б изведена из класе А, у генеричку листу објеката класе Б, могу се убацити и објекти класе А и објекти класе Б
3. Објекат типа Б се може пренети као аргумент методу на место параметра типа А уколико је Б класа наследница од А.
4. Када се у класи Б која је изведена из класе А изврши предефинисање (override) наслеђене методе, онда су у објекту класе Б доступне обе верзије ове методе (исто као код преоптеређивања метода – overload)

2

233. Дат је код програма у програмском језику C#. Код садржи објекте две класе у којима је дефинисан метод **ToString()**. Анализирати код датог програма и одредити који од датих исказа су тачни.

```
namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 Object a = new Klasa();
 Object obj = new Object();
 Console.WriteLine(a);
 Console.WriteLine(obj);
 }
 }
 class Klasa{
 int x;
 public override string ToString() { return "x у А је " + x; }
 }
}
```

2

Заокружити бројеве испред очекиваних одговора:

1. Програм има грешку, јер наредбу **Console.WriteLine(a)** треба заменити наредбом **Console.WriteLine(a.ToString())**.
2. Приликом извршавања наредбе **Console.WriteLine(a)**, програм позива се метод **ToString()** наслеђен из класе **Object**.
3. Приликом извршавања наредбе **Console.WriteLine(a)**, програм позива метод **ToString()** из класе **A**.
4. Приликом извршавања наредбе **Console.WriteLine(obj)**, програм позива метод **ToString()** из класе **Object**.

234. Дат је код програма у програмском језику C# којим су дефинисане четири класе: `class Program` која садржи `Main(String[] args)` методу, `class MasterStudent`, `class Student` и `class Osoba`. Приликом превођења овог кода компајлер јавља грешку. Заокружити бројеве испред понуђених исказа који описују разлоге настајања грешке приликом компајлирања.

```

class Program {
 static void Main(string[] args) {
 m(new MasterStudent());
 m(new Student());
 m(new Osoba());
 m(new Object());
 }
 public static void m(Student x) {
 Console.WriteLine(x.ToString());
 }
}
class MasterStudent : Student{ }
class Student: Osoba{
 public string ToString(){ return "Student"; }
}
class Osoba: Object{
 public string ToString(){ return "Osoba"; }
}

```

2

1. Грешка је у позиву `m(new MasterStudent())`.
2. Грешка је у позиву `m(new Student())`.
3. Грешка је у позиву `m(new Osoba())`.
4. Грешка је у позиву `m(new Object())`.

235. У програмском језику C# дата је декларација две класе: **KlasaA** и **KlasaB** која наслеђује класу **KlasaA**. Анализирати дате класе и проценити који од понуђених исказа су тачни.

```

namespace TestPrimer {
 class Program {
 static void Main(string[] args) {
 KlasaB b = new KlasaB();
 b.Print();
 }
 }
 class KlasaA {
 string s;
 public KlasaA(string s) { this.s = s; }
 public void Print() { Console.WriteLine(this.s); }
 }
 class KlasaB :KlasaA{ }
}

```

2

Заокружити бројеве испред очекиваних одговора:

1. Програм има грешку, јер **KlasaB** нема подразумевани конструктор **KlasaB()**.
2. Програм има грешку јер **KlasaB** има подразумевани конструктор, док родитељска **KlasaA** нема такав конструктор. Програм би радио без грешке уколико би се уклонио конструктор са параметрима из **KlasaA**.
3. Програм има грешку која се може отклонити уколико би се у **KlasaA** експлицитно додао конструктор без параметара **KlasaA()**.
4. Програм нема грешку, извршава се, али се на конзоли ништа не исписује јер је поље `s` добило подразумевану вредност **String.Empty**

236. Дат је код на C#-у којим су креиране три класе у ланцу наслеђивања. Имајући у виду класификаторе приступа пољима класа, заокружити бројеве испред поља која ће бити видљива унутар класе Sin:

```
public class Deda {
 private double penzija;
 protected string adresa;
 private int godinaRodjenja;
 public string ime;
}
public class Otac: Deda {
 private double plata;
 protected string struka;
 private string brojRacuna;
}
public class Sin: Otac {
 public int razred;
}
```

|                                                                                                               |   |
|---------------------------------------------------------------------------------------------------------------|---|
| 1. penzija<br>2. adresa<br>3. ime<br>4. plata<br>5. struka<br>6. razred<br>7. godinaRodjenja<br>8. brojRacuna | 2 |
|---------------------------------------------------------------------------------------------------------------|---|

237. Дат је код на C#-у којим су креиране три класе у ланцу наслеђивања. Имајући у виду класификаторе приступа пољима класа, заокружити бројеве испред поља која ће бити видљива унутар класе Sin:

```
public class Deda {
 private double penzija;
 protected string adresa;
 public string ime;
}
public class Otac: Deda {
 private double plata;
 protected string struka;
}
public class Sin: Deda {
 public int razred;
}
```

|                                                                         |     |
|-------------------------------------------------------------------------|-----|
| 1. penzija<br>2. adresa<br>3. ime<br>4. plata<br>5. struka<br>6. razred | 1,5 |
|-------------------------------------------------------------------------|-----|

238. Дат је код на C#-у којим су креиране три класе у ланцу наслеђивања. Унутар сваке класе декларисан је по један private, public и protected атрибут. У методи **Main()** класе **Program** креiran је објекат s класе **Sin** (**Sin s = new Sin();**) Заокружити бројеве испред поља која ће бити видљива у креiranом објекту s класе Sin:

```
public class Deda {
 private double penzija;
 protected string adresa;
 public string ime;
}
public class Otac: Deda {
 private double plata;
 protected string firma;
 public string struka;
}
public class Sin: Otac {
 private double prosek;
 protected int razred;
 public string skola;
}
```

|                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------|-----|
| 1. penzija<br>2. adresa<br>3. ime<br>4. plata<br>5. struka<br>6. firma<br>7. prosek<br>8. razred<br>9. skola | 1,5 |
|--------------------------------------------------------------------------------------------------------------|-----|

239. Дат је код програма у програмском језику C# који декларише два објекта типа **Pravougaonik** и један објекат типа **Krug**. Класе Pravougaonik и Krug наслеђују апстрактну класу **Figura** и обе класе имају методу **ToString**. Код програма треба допунити наредбом која ће исписивати низ **Figura[]** декларисан у коду програма.

```

1. public abstract class Figura {
2. protected static int brojFigura=0;
3. public static int BrojFigura {get{return brojFigura;}}
4. public Figura() { brojFigura++; }
5. public abstract double povrsinaFigure();
6. public override string ToString(){return "Figura ";}
7. }
...
1. Pravougaonik pr1 = new Pravougaonik();
2. Pravougaonik pr2 = new Pravougaonik(3, 2);
3. Krug krl = new Krug(2);
4.
5. Console.WriteLine("Broj figura=" + Figura.BrojFigura + "\n");
6. Figura[] niz = new Figura[Figura.BrojFigura];
7. niz[0] = pr1; niz[1] = pr2; niz[2] = krl;
8.
9. Console.WriteLine("Prikaz figura:");
10.

```

2

Заокружити бројеве испред одговора који представљају могуће начине приказа низа фигура:

1. `foreach(Figura el in niz) Console.WriteLine(el);`
2. `for(int i=0; i<niz.Length; i++) Console.WriteLine(niz[i]);`
3. `foreach (el in niz) Console.WriteLine(el);`
4. `foreach (int i in niz) Console.WriteLine(niz[i]);`

240. Дати су изкази који се односе на правила писања try-catch-finally блокова за руковање изузетима. Заокружити бројеве испред исказа који су тачни:

1. Блок try мора имати бар један catch блок
2. Блок try може имати више catch блокова
3. Ако блок try има више catch блокова, изузетак основне Exception класе мора се хватати у првом catch блоку
4. Ако блок try има више catch блокова, битан је редослед њиховог писања
5. Блок try мора имати бар један finally блок
6. Блок try не сме да има више catch блокова

1,5

### Допуните следеће реченице и табеле

241. Допунити дате изјаве уносом кључне речи која недостаје:

Преко јавних својстава (public property) може се остварити приступ приватним члановима класе.

1

Постављање вредности приватног поља врши се наредбом \_\_\_\_\_, док се читање вредности приватног поља врши наредбом \_\_\_\_\_ у оквиру јавног својства.

242. Са леве стране, у првој колони су наведени елементи, а у другој класе класификатори приступа елементима. Допуните реченицу која се односи на подразумевано право приступа наведеним елементима.

На линију у уписати одговарајуће појмове наведене у колонама:

|                     |                          |                                      |
|---------------------|--------------------------|--------------------------------------|
| 1. Класа            | 1. private               | Сви елементи наведени у првој колони |
| 2. Полеје класе     | 2. protected             | са лева имају подразумевани          |
| 3. Својство у класи | 3. public                | класификатор приступа _____,         |
| 4. Метод класе      | 4. internal              | осим _____ где је                    |
| 5. Конструктор      | 5. protected<br>internal | подразумевани приступ _____.         |

2

243. Допунити дате изјаве уносом кључне речи која недостаје:

Метод дефинисан у родитељској класи, у класама наследницима може бити редефинисан или сакривен.

Да би се омогућило **редефинисање** методе, при дефиницији у родитељској класи, испред ознаке повратног типа метода наводи се кључна реч **virtual**, **abstract** или **override**, док у изведеном класи испред ознаке повратног типа треба навести кључну реч \_\_\_\_\_.

1

**Сакривање** методе родитељске класе врши се тако што се у изведеном класи испред ознаке повратног типа наведе кључна реч \_\_\_\_\_.

244. На програмском језику C# дефинисане су две класе:

```
public class Racun {
 public virtual int Uvecaj() { return 10; }
}
public class Dinarski: Racun {
 public override int Uvecaj() { return 20 * base.Uvecaj(); }
}
public class Devizni : Dinarski {
 public override int Uvecaj() { return 50 + base.Uvecaj(); }
}
```

Унутар функције Main, креирана су три објекта ових класа на следећи начин:

3

```
Racun r = new Racun();
Racun rDin = new Dinarski();
Racun rDev = new Devizni();
```

Анализирати код и на предвиђене линије уписати шта метод Uvecaj() враћа при позиву из наведених објеката:

r.Uvecaj(); \_\_\_\_\_  
rDin.Uvecaj(); \_\_\_\_\_  
rDev.Uvecaj(); \_\_\_\_\_

245. На програмском језику C# дефинисане су две класе:

```
public class Racun {
 public virtual int Uvecaj() { return 10; }
}
public class Dinarski: Racun {
 public override int Uvecaj() { return 20 * base.Uvecaj(); }
}
public class Devizni : Racun {
 public override int Uvecaj() { return 50 + base.Uvecaj(); }
}
```

Унутар функције Main, креирана су три објекта ових класа на следећи начин:

3

```
Racun r = new Racun();
Racun rDin = new Dinarski();
Racun rDev = new Devizni();
```

Анализирати код и на предвиђене линије уписати шта метод Uvecaj() враћа при позиву из наведених објеката:

r.Uvecaj(); \_\_\_\_\_  
rDin.Uvecaj(); \_\_\_\_\_  
rDev.Uvecaj(); \_\_\_\_\_

246. На програмском језику C# дефинисане су две класе:

```
public class Roditelj {
 public virtual void Poruka1(){ Console.WriteLine("R1"); }
 public void Poruka2(){ Console.WriteLine("R2"); }
}
public class Dete: Roditelj {
 public override void Poruka1(){ Console.WriteLine("D1"); }
 public new void Poruka2(){ Console.WriteLine("D2"); }
}
```

Унутар функције Main креирана су два објекта ових класа на следећи начин:

4

```
Dete x = new Dete();
Roditelj y = new Dete();
```

Проценити ефекат извршења наведених позива и на предвиђене линије уписати шта ће се видети на стандардном излазу извршењем позваних метода:

x.Poruka1(); \_\_\_\_\_  
x.Poruka2(); \_\_\_\_\_  
y.Poruka1(); \_\_\_\_\_  
y.Poruka2(); \_\_\_\_\_

247. Дате су започете изјаве које се односе на делове кода за обраду изузетака. Довршити започете реченице:

Наредбе које се извршавају у случају настанка грешке, стављају се унутар блока \_\_\_\_\_

1,5

Наредбе које се извршавају и ако се деси и ако се не деси грешка, стављају се унутар блока \_\_\_\_\_

Наредбе које могу узводити грешку стављају се унутар блока \_\_\_\_\_

248. На програмском језику C# дефинисане су две класе:

```
public class KlasaA {
 public virtual int Metod() { return 10; } }
public class KlasaB : KlasaA {
 public override int Metod() { return 20; } }
public class KlasaC : KlasaB {
 public new int Metod() { return 30; } }
```

Креирани су објекти ових класа и из њих позвана метода **Metod()**. На предвиђене линије уписати шта метод **Metod()** враћа при позиву из наведених објеката:

3

KlasaA a = new KlasaA(); a.Metod() враћа вредност \_\_\_\_\_  
 KlasaB b = new KlasaB(); b.Metod() враћа вредност \_\_\_\_\_  
 KlasaA bb = new KlasaB(); bb.Metod() враћа вредност \_\_\_\_\_  
 KlasaC c = new KlasaC(); c.Metod() враћа вредност \_\_\_\_\_  
 KlasaB cc = new KlasaC(); cc.Metod() враћа вредност \_\_\_\_\_  
 KlasaA ccc = new KlasaC(); ccc.Metod() враћа вредност \_\_\_\_\_

249. На програмском језику C# дефинисана је структура **SVreme** и класа **CVreme**, а затим креирање по две променљиве типа **SVreme** и **CVreme**:

```
public struct SVreme {
 public int sat;
 public int min;
 public SVreme(int h,int m) {
 sat = h; min = m;
 }
}
public class CVreme {
 public int sat;
 public int min;
 public CVreme(int h,int m) {
 sat = h; min = m;
 }
}
SVreme sv1 = new SVreme(12,15);
SVreme sv2 = sv1;
CVreme cv1 = new CVreme(12,15);
CVreme cv2 = cv1;
```

2

По извршењу следећих наредби доделе вредности пољима променљивих **st1** и **cv1**, одредити и на одговарајућу линију уписати вредности које ће имати одговарајућа поља променљивих **st2** и **cv2**.

sv1.sat = 24; sv2.sat = \_\_\_\_\_;  
 cv1.sat = 24; cv2.sat = \_\_\_\_\_;

### У следећим задацима уредите и повежите појмове према захтеву

250. Са леве стране су наведени делови/елементи класе, а са десне стране улоге поједињих класних елемената. На линију испред описа улоге унети редни број под којим је наведен одговарајући елеменат класе:

- |                        |       |                                    |
|------------------------|-------|------------------------------------|
| 1. поље (атрибут)      | _____ | Опис функционалности објекта класе |
| 2. деструктор          | _____ | Контрола приступа пољима класе     |
| 3. конструктор         | _____ | Опис особина објекта класе         |
| 4. метод               | _____ | Креирање објекта класе             |
| 5. својство / property | _____ | Уништавање објекта класе           |

2,5

|      |                                                                                                                                                                                                                                                                                      |                                                                                                                        |     |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|-----|
| 251. | Са десне стране су наведене су области видљивости поједињих елемената класе, а са леве стране класификатори приступа којима се врши контрола области видљивости. На линију испред описа области видљивости унети редни број под којим је наведен одговарајући класификатор приступа: |                                                                                                                        |     |
| 1.   | private                                                                                                                                                                                                                                                                              | видљив унутар класе у којој је дефинисан, као и унутар изведенih класа                                                 | 2   |
| 2.   | public                                                                                                                                                                                                                                                                               | видљив само унутар класе у којој је дефинисан                                                                          |     |
| 3.   | protected                                                                                                                                                                                                                                                                            | видљив унутар пројекта у коме је дефинисан                                                                             |     |
| 4.   | internal                                                                                                                                                                                                                                                                             | видљив и ван своје класе у којој је дефинисан                                                                          |     |
| 252. | Са леве стране дате су врсте променљивих у програмском језику C#, а са десне декларације променљивих. На линију испред декларације променљиве унети број којим је означена одговарајућа врста променљиве:                                                                            |                                                                                                                        |     |
| 1.   | Променљива вредносног типа                                                                                                                                                                                                                                                           | <code>const double eps0 = 8.85E-12</code>                                                                              | 2,5 |
| 2.   | Променљива референтног типа                                                                                                                                                                                                                                                          | <code>int x = 10;</code>                                                                                               |     |
| 3.   | Именована константа                                                                                                                                                                                                                                                                  | <code>int[] a = new int[7];</code><br><code>int[,] b=new int[10,10];</code><br><code>int[] c = {1,2,3,4,5,6,7};</code> |     |
| 253. | Са леве стране дате су кључне речи које одређују типове класа, а са десне су описи класа. На линију испред описа уписати редни број под којим је наведен одговарајући тип класе:                                                                                                     |                                                                                                                        |     |
| 1.   | abstract                                                                                                                                                                                                                                                                             | Класа која се простира у више фајлова                                                                                  |     |
| 2.   | sealed                                                                                                                                                                                                                                                                               | Класа садржи само декларације метода, али не и дефиницију (тело) методе                                                | 2   |
| 3.   | partial                                                                                                                                                                                                                                                                              | Класа која се не може инстанцирати                                                                                     |     |
| 4.   | interface                                                                                                                                                                                                                                                                            | Класа из које се не може наслеђивати                                                                                   |     |
| 254. | Са леве стране наведене су врсте грешака које се могу десити у коду, а са десне описи тих грешака. На линију испред описа уписати редни број под којим је наведен одговарајући тип грешке:                                                                                           |                                                                                                                        |     |
| 1.   | Синтаксне грешке<br>(syntax error)                                                                                                                                                                                                                                                   | Откривају се у фази компилације и најлакше су за кориговање                                                            | 1,5 |
| 2.   | Логичке грешке<br>(logical error)                                                                                                                                                                                                                                                    | Не морају, али се могу десити током извршења и тада доводе до „пуцања“ програма                                        |     |
| 3.   | Грешке у време извршења<br>(run-time error)                                                                                                                                                                                                                                          | Програм функционише, нема „пуцања“, али се добијају погрешни резултати                                                 |     |

## ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК ЈАВА

### У следећим задацима заокружите број испред траженог одговора

255. Дати су типови променљивих у програмском језику Јава. Одредити како се назива променљива која је дефинисана унутар неког метода.  
Заокружити број испред очекиваног одговора:
1. Глобална променљива
  2. Статичка променљива
  3. Блоковска променљива
  4. Локална променљива
256. Дате су програмске јединице у програмском језику Јава. Одредити која програмска јединица представља шаблон за конструисање објекта истог типа.  
Заокружити број испред очекиваног одговора:
1. Класа
  2. Пакет
  3. Нестатичка метода
  4. Променљива
  5. Статичка метода
257. Одредити какви могу бити чланови класе (поља и методе) у програмском језику Јава.  
Заокружити број испред очекиваног одговора:
1. Локални и глобални
  2. Процедурални и непроцедурални
  3. Статички (класни) и нестатички (објектни)
  4. Спољашњи и унутрашњи
258. Дат је код програма у програмском језику Јава. Анализирати дати код и проценити његову тачност. Заокружити број испред понуђеног тачног исказа:
- ```
public class Test {  
    int x;  
    public Test(String s) {  
        System.out.println("Test");  
    }  
    public static void main(String[] args) {  
        Test t = null;  
        System.out.println(t.x);  
    }  
}
```
1. Програм има грешку јер променљива x није иницијализована.
 2. Програм има грешку јер класа Test нема подразумевани конструктор.
 3. Програм има грешку јер се у некој класи не може декларисати променљива типа те исте класе, као што је то овде случај са променљивом t.
 4. Програм има грешку јер променљива t није иницијализована и има вредност **null** у моменту када се приказује поље **t.x**.
 5. Програм нема грешака и нормално се извршава, не приказујући ништа на екрану.

259. Дати код програма у програмском језику Јава састоји се од две класе у једној датотеци. Анализирати дати код и проценити његову тачност.

```
public class Test {
    public static void main(String[] args) {
        Klasa a = new Klasa();
        a.n++;
    }
}
class Klasa {
    int n;
    private Klasa() {}
}
```

Заокружити број испред тачног исказа:

2

1. Програм има грешку јер класа **Klasa** има приватни конструктор и приватно поље **n**.
2. Програм има грешку јер класа **Klasa** има празан подразумевани конструктор.
3. Програм има грешку јер променљива **n** није иницијализована.
4. Програм нема грешака и нормално се извршава

260. Дат је код програма у програмском језику Јава:

```
public class TestPrimer {
    public static void main(String[] args) {
        System.out.println(fun(17));
    }
    public int fun(int n) { return n; }
    public void fun(int n) { System.out.println(n); }
}
```

Анализирати код и заокружити број испред очекиваног одговора:

2

1. Програм има грешку, јер се не може одредити коју верзију преоптерећеног метода **fun()** треба позвати.
2. Програм има грешку, јер је друга верзија преоптерећеног метода **fun()** дефинисана али се нигде не позива.
3. Програм се нормално извршава и приказује 17 једанпут.
4. Програм се нормално извршава и приказује 17 двапут.

261. У програмском језику Јава дата је декларација низа:

```
int k;
int[ ] brojevi = {5, 12, 37, 7, 27, 33, 36};
```

На основу дате декларације одредити шта је резултат позива

k=Arrays.binarySearch(brojevi, 37);

Заокружити број испред очекиваног одговора:

2

1. k=0, јер метод **binarySearch** прво изврши сортирање низа у опадајућем редоследу, па онда тражи задату вредност
2. метод **binarySearch** баца изузетак **увек** када је низ неуређен и програм „пуца“
3. k=2, јер се тражени елеменат налази на позицији 2
4. k добија неочекивану вредност јер низ мора бити сортиран у растућем поретку пре позива методе **binarySearch**
5. k=6, јер метод **binarySearch** прво изврши сортирање низа у растућем редоследу, па онда тражи задату вредност

262. У програмском језику Јава декларисане су статичке променљиве логичког, нумеричког и класног типа.

```
public class Test {
    static boolean x;
    static int y;
    static NekaKlasa r;
    public static void main(String[] args) { ... }
}
```

Одредити које аутоматске почетне (default) вредности декларисана поља имају унутар методе Main. Понуђени одговори су наведени у наглашеном редоследу. Заокружити број испред очекиваног одговора:

	x	y	r
1.	true	1	null
2.	false	0	null
3.	true	0	null
4.	false	1	null
5.	false	0	void

1

263. Дат је код програма у програмском језику Јава и састоји се од две класе у једној датотеци. Анализирати дати код и проценити која се вредност поља **b.n** приказује првом наредбом println (ред седам датог кода) приликом извршавања овог програма. Заокружити број испред очекиваног одговора:

```
1. public class Test {
2.     public static void main(String[] args) {
3.         int k=0;
4.         Brojac b=new Brojac();
5.
6.         for(int i=0;i<100; i++) inc(b,k);
7.         System.out.println("b.n = " + b.n);
8.         System.out.println("k = " + k);
9.     }
10.    public static void inc(Brojac b, int k){
11.        b.n++;
12.        k++;
13.    }
14.}
15. class Brojac{
16.     int n;
17.     public Brojac(int n){ this.n=n; }
18.     public Brojac(){ this.n=1; }
19.}
```

2

1. b.n = 101
2. b.n = 100
3. b.n = 99
4. b.n = 0
5. b.n = 1

264. У програмском језику Јава дефинисана је метода са аргументом низовног типа. Одредити шта се тачно преноси том методу. Заокружити број испред очекиваног одговора:

1. Копија датог низа.
2. Копија првог елемента датог низа.
3. Референца на дати низ.
4. Дужина датог низа

1

265. Дата је наредба кода у програмском језику Јава која представља дефиницију низа **niz**. Проценити која од доле наведених дефиниција низа је коректна.

Заокружити број испред очекиваног одговора:

1. int[] niz = new int[20];
2. int[] niz = new int(20);
3. int niz = new int[20];
4. int niz() = new int[20];

1

266. Дат је код програма у програмском језику Јава који формира и штампа елементе низа **a**. Анализирати дати код и проценити шта ће се догодити након његовог извршавања.

```
public class TestPrimer {
    public static void main(String[] args) {
        int[] a = new int[5];
        int i;
        for (i = 0; i < a.length; i++) a[i] = i;
        System.out.print(a[i] + " ");
    }
}
```

2

Заокружити број испред очекиваног одговора:

1. Програм приказује бројеве 0 1 2 3 4 на екрану.
2. Програм има грешку, јер ће у последњој наредби **print** метода **main** покушати приступ непостојећем елементу **a[5]**.
3. Програм приказује број 5 на екрану.
4. Програм има грешку јер променљива **i** у последњој наредби **print** у методу **main** неће имати дефинисану вредност.

267. Дат је код програма у програмском језику Јава. Анализирати програм и одредити шта се приказује на екрану као резултат његовог извршавања.

```
public class TestPrimer {
    public static void main(String[] args) {
        int[] x = {0, 1, 2, 3, 4};
        inc(x);
        int[] y = {0, 1, 2, 3, 4};
        inc(y[0]);
        System.out.println(x[0] + " " + y[0]);
    }
    public static void inc(int[] a) {
        for (int i = 0; i < a.length; i++) a[i]++;
    }
    public static void inc(int n) { n++; }
}
```

2

Заокружити број испред очекиваног одговора:

1. Порука о грешци
2. 1 0
3. 2 2
4. 2 1
5. 1 1

268. Заокружити број испред исправно написане наредбе кода у програмском језику Јава којом се копија низа **x** додељује низу **y**:

1. **y = Arrays.copyOf(x, x.length);**
2. **y = Arrays.copyOf(x);**
3. **Arrays.copyOf(y, a, a.length);**
4. **Arrays.copyOf(x, y);**

1

269. Дат је програм у програмском језику Јава у ком је дефинисан низ и метода **prosek** која израчунава средњу вредност низа. Анализирати дати код и проценити шта ће се десити након његовог извршавања.

```
public class TestPrimer {
    public static void main(String[] args) {
        double[] niz = {1.0, 2.0, 3.0};
        System.out.println(prosek(niz));
        System.out.println(prosek(1, 2, 2, 1, 4));
        System.out.println(prosek(new double[]{1, 2, 3}));
        System.out.println(prosek(1.0, 2.0, 2.0, 1.0));
    }
    public static double prospek (double... brojevi) {
        double suma = 0;
        for (double elem : brojevi) suma += elem;
        return suma / brojevi.length;
    }
}
```

Заокружити број испред очекиваног одговора:

1. Програм има грешку у првој наредби `.println`, јер је неправилан позив функције **prosek(niz)**.
2. Програм има грешку у другој наредби `.println`, јер је неправилан позив функције **prosek(1, 2, 2, 1, 4)**.
3. Програм има грешку у трећој наредби `.println`, јер је неправилан позив функције **prosek(new double[]{1, 2, 3})**.
4. Програм има грешку у четвртој наредби `.println`, јер је неправилан позив функције **prosek(1.0, 2.0, 2.0, 1.0)**.
5. Програм се извршава без грешке и просек датих бројева се тачно израчунава.
6. Програм се извршава без грешке, али се просек датих бројева не израчунава тачно.

2

2

270. У програмском језику Јава дата је декларација променљиве једне стринг и једне целебројне променљиве, као и део кода:

```
String str = "Primer";
int broj=66;
System.out.println(str+broj+65);
System.out.println(65+broj+str);
```

Анализирати код и проценити шта ће се приказати на екрану након његовог извршења.
Заокружити број испред очекиваног одговора:

1. Primer6665
2. Primer6665
131Primer
3. Primer131
6665Primer
4. PrimerBA
BAPrimer

271. Дат је део кода који је написан на Јава програмском језику. Анализирати код и одредити шта ће бити на излазу.

```
class Test
{
    public static void main(String [] args) {
        Test p = new Test();
        p.start();
    }
    void start() {
        boolean b1 = false;
        boolean b2 = fix(b1);
        System.out.println(b1 + " " + b2);
    }
    boolean fix(boolean b1) {
        b1 = true;
        return b1;
    }
}
```

Заокружи број испред тачног одговора:

1. true true
2. false true
3. true false
4. false false

2

272. Дата је наредба кода у програмском језику Јава која представља позив методе класе Arrays. Одредити којим од датих позива се сортира низ **brojevi** типа int[].

Заокружити број испред очекиваног одговора:

1. Arrays(brojevi)
2. Arrays.sorts(brojevi)
3. Arrays.sortArray(brojevi)
4. Arrays.sort(brojevi)

1

273. На програмском језику Јава дефинисане су класе:

```
class A {
    public int Metod() { return 10; }
}
class B extends A {
    public final int Metod() { return super.Metod() + 20; }
}
class C extends B {
    public int Metod() { return super.Metod() + 30; }
```

Унутар функције Main, креирана су три објекта ових класа и из сваког од њих извршен позив методе **Metod()**

2

Заокружити број испред понуђеног одговора који по вашем мишљењу представља сценарио који ће се десити при покретању програма:

1. Метод позван из класе А враћа вредност 10, из класе В 10, а из класе С 30
2. Метод позван из класе А враћа вредност 10, из класе В 30, а из класе С 60
3. Компилатор јавља грешку јер родитељска класа В не може да има final чланице уколико се из ње изводе нове класе
4. Компилатор јавља грешку јер у класи С метод не може бити редефинисан

274. У програмском језику Јава, по дефиницији метод може да врати вредност, али и не мора. Изабрати службену реч која се користи за означавање типа резултата методе када дефинишемо метод који НЕ враћа вредност. Заокружити број испред очекиваног одговора.

1. void
2. return
3. public
4. static

1

275. У програмском језику Јава дат је метод **PrintPoruka** који је дефинисан на следећи начин:

```
void PrintPoruka(String poruka, int n) {
    while (n > 0) {
        System.out.print(poruka);
        n--;
    }
}
```

Одредити шта ће бити исписано на екрану након позива метода **PrintPoruka("b",3)**. Заокружити број испред очекиваног одговора:

1

1. На екрану се приказује bbbbb
2. На екрану се приказује bbbb
3. На екрану се приказује bbb
4. На екрану се приказује bb
5. На екрану се приказује b

276. У програмском језику Јава дат је метод **PrintPoruka** који је дефинисан на следећи начин:

```
void PrintPoruka(String poruka, int n) {
    while (n > 0) {
        n--;
        System.out.print(poruka);
    }
}
```

Израчунати вредност променљиве **m** након извршења следећег програмског фрагмента:

1

```
int m = 2;
PrintPoruka("Java!!!", m);
```

Заокружити број испред очекиваног одговора:

1. m =0
2. m =1
3. m =2
4. m =3

277. Дат је код у програмском језику Јава, који дефинише рекурзивни метод. Анализирати код и одредити резултат извршавања задатог метода.

```
public long fun(int n) {
    return n * fun(n - 1);
}
```

Заокружити број испред очекиваног одговора:

2

1. Резултат позива **fun(3)** је 1.
2. Резултат позива **fun(3)** је 2.
3. Резултат позива **fun(3)** је 6.
4. Позив **fun(3)** изазива грешку јер производи бесконачан ланац позива истог метода **fun(...)**.

278. Дат је код у програмском језику Јава, који дефинише рекурзивни метод. Анализирати код и одредити резултат који ће се приказати на екрану.

```
public class TestPrimer {
    public static void main(String[] args) {
        fun(2);
    }
    public static void fun(int n) {
        while (n > 1) {
            System.out.print((n - 1) + " ");
            fun(n - 1);
        }
    }
}
```

2

Заокружити број испред очекиваног одговора:

1. Програм на екрану не приказује ништа
2. Програм на екрану приказује 1 2 3
3. Програм на екрану приказује 3 2 1.
4. Програм на екрану бесконачно приказује 1 1 1 1 1
5. Програм на екрану бесконачно приказује 2 2 2 2 2

279. Дат је код у програмском језику Јава који дефинише рекурзивни метод. Анализирати код и одредити резултат извршавања задатог метода.

Заокружити број испред очекиваног одговора:

```
public static int fun(int n) {
    if (n == 1) return 1;
    else return n + fun(n - 1);
}
```

2

1. Позивом fun (4) се исти метод fun () позива још 2 пута.
2. Позивом fun (5) се исти метод fun () позива још 4 пута.
3. Позивом fun (4) се исти метод fun () позива још 4 пута.
4. Позивом fun (5) се исти метод fun () позива још 6 пута.

280. У програмском језику Јава дат је рекурзивни метод који проверава да ли је неки стринг палиндром. Да би код био комплетиран потребно је допунити трећи ред са условом за if наредбум

```
1. public static boolean palindrom(String s) {
2.     if (s.length() <= 1) return true; //bazni slučaj
3.     else if (_____) return false;
4.     else return palindrom(s.substring(1, s.length() - 1));
5. }
```

2

Заокружити број испред очекиваног одговора:

1. s.charAt(0) != s.charAt(s.length() - 1)
2. s.charAt(0) != s.charAt(s.length())
3. s.charAt(1) != s.charAt(s.length() - 1)
4. s.charAt(1) != s.charAt(s.length())

281. Заокружити број испред појма чији је опис дат у тексту питања.

Процес скривања неких података о објекту од корисника и пружања само неопходних података, зове се...

1. Полиморфизам
2. Наслеђивање
3. Енкапсулација
4. Апстракција

1

282. У програмском језику Јава дат је рекурзивни метод који проверава да ли је неки стринг палиндром. Да би код био комплетиран потребно је допунити седми ред.

```

1. public static boolean palindrom(String s) {
2.     return palindrom(s, 0, s.length() - 1);
3. }
4. public static boolean palindrom(String s,int levi, int desni){
5.     if (desni <= levi) return true; // bazni slucaj
6.     else if (s.charAt(levi) != s.charAt(desni)) return false;
7.     else return _____;
8. }
```

2

Заокружити број испред очекиваног одговора:

1. *palindrom(s)*
2. *palindrom(s, levi, desni)*
3. *palindrom(s, levi+1, desni-1)*
4. *palindrom(s, levi+1, desni)*
5. *palindrom(s, levi, desni-1)*

283. У програмском језику Јава дат је рекурзивни метод за сортирање низа рационалних бројева у опадајућем редоследу. Да би код био комплетиран потребно је допунити други ред помоћу једног од понуђених одговора.

```

1. public static void sortiranje(double[] niz) {
2.     _____;
3. }
4. public static void sortiranje(double[] niz, int kraj) {
5.     if (kraj > 0) {
6.         int imin = 0;
7.         double min = niz[0];
8.         for (int i = 1; i <= kraj; i++) {
9.             if (niz[i] < min) {
10.                 min = niz[i];
11.                 imin = i;
12.             }
13.             niz[imin] = niz[kraj];
14.             niz[kraj] = min;
15.             sortiranje(niz, kraj - 1);
16.         }
17.     }
}
```

2

Заокружити број испред очекиваног одговора:

1. *sortiranje(niz)*
2. *sortiranje(niz, niz.length)*
3. *sortiranje(niz, niz.length+1)*
4. *sortiranje(niz, niz.length-1)*

284. У програмском језику Јава дат је рекурзивни метод за бинарно претраживање сортираног целобројног низа. Да би код био комплетиран потребно је допунити девети ред (означен линијом) помоћу понуђеног одговора.

```

1. public static int traziBroj(int[] niz, int broj) {
2.     return traziBroj(niz, broj, 0, niz.length - 1);
3. }
4. public static int traziBroj(int[] niz, int broj, int levi, int desni) {
5.     if (levi > desni) return -1; // broj nije nadjen u nizu
6.     int sredina = (levi+desni)/2;
7.     if (broj<niz[sredina]) return traziBroj(niz, broj, levi, sredina-1);
8.     else if(broj>niz[sredina]) return _____;
9.     else return sredina;
10. }
```

2

Заокружити број испред очекиваног одговора:

1. *traziBroj(niz, broj, sredina + 1, levi)*
2. *traziBroj(niz, broj, sredina - 1, levi)*
3. *traziBroj(niz, broj, desni, sredina + 1)*
4. *traziBroj(niz, broj, sredina + 1, desni)*

285. Одредити која поља су заједничка и јединствена за све креиране објекте неке класе дефинисане у објектно оријентисаном програмском језику Јава.

Заокружити број испред очекиваног одговора:

1. Јавна
2. Приватна
3. Објектна
4. Инстанцна
5. Статичка

1

286. У програмском језику Јава класа може да садржи статичка и не-статичка (инстанцна) поља. Дате су изјаве које се односе на статичка поља класе и међу њих је уметнута једна изјава која се односи на не-статичка (инстанцна) поља класе.

Заокружити број испред изјаве која се односи на не-статичка поља класе:

1. Поље које се може користити без конструисања иједног објекта те класе
2. Поље које има исту вредност за све креиране објекте неке класе
3. Поље чија вредност може разликовати за сваки појединачни објекат неке класе
4. Поље које се може користити унутар статичких метода класе, као и унутар метода инстанце

1

287. Дата је дефиниција класе у програмском језику Јава. Проценити где у дефиницији класе (исpred које методе) треба заменити знакове ??? службеном речју **static**.

```

1. public class Test {
2.     private int broj;
3.
4.     public ??? int kvadrant(int n) { return n * n; }
5.     public ??? int getBroj() { return broj; }
6. }
```

2

Заокружити број испред тачне изјаве:

1. Метода **kvadrant** МОРА да буде статичка, док метода **getBroj** може и не мора.
2. Обе методе морају бити статичке.
3. Ни једна од дефинисаних метода није статичка.
4. Метода **getBroj** НЕ СМЕ да буде статичка, док метода **kvadrant** може и не мора.

288. Дата је наредба кода у програмском језику Јава која дефинише константу MAX_CENA као чланицу неке класе. Проценити који од понуђених начина дефинисања је исправан.

Заокружити број испред очекиваног одговора:

1. **final static** MAX_CENA = 100.00;
2. **final static float** MAX_CENA = 99.50;
3. **static double** MAX_CENA = 100.50;
4. **const double** MAX_CENA = 100.99;
5. **final static double** MAX_CENA = 100.99;

1

289. Дат код програма у програмском језику Јава. Анализирати дати код и проценити његову тачност. Заокружити број испред очекиваног одговора:

```
public class proba {
    public static void main(String[] args) {
        int n = 2;
        fun(n);
        System.out.println("n je " + n);
    }
    void fun(int n) { n++; }
}
```

2

1. Програм има грешку, јер метод fun() не враћа ниједну вредност.
2. Програм има грешку, јер метод fun() није дефинисан да буде статички.
3. Програм приказује 1 на екрану.
4. Програм приказује 2 на екрану.
5. Програм приказује 3 на екрану.

290. Дата је дефиниција класе у програмском језику Јава и састоји се од два конструктора, методе и поља x и y. У шестом реду написати конструктор копије класе Point.

```
1. public class Point {
2.     private double x,y;
3.     public Point( ) { x=0; y=0; }
4.     public void set(double xx,double yy) { x=xx; y=yy; }
5.     public Point(Point p) {
6.         _____ //Odgovor
7.     }
8. }
```

2

Заокружити број испред очекиваног одговора:

1. this(p.x,p.y);
2. this(p);
3. set(p);
4. set(p.x,p.y);

291. У програмском језику Јава, метод equals(...) је метод instance класе Object којим се проверава да ли је објекат из кога се метод позива једнак неком задатом објекту. Овај метод се може надјачати (override-овати) у наслеђеним класама. Одредити заглавље овог метода у класи String у којој би метод био надјачан.

Заокружити број испред понуђеног тачног одговора:

1. public boolean equals(String s)
2. public new boolean equals(String s)
3. public boolean equals(Object o)
4. public static boolean equals(Object o)
5. public boolean equals(String s1, String s2)

2

292. У програмском језику Јава, метод **equals()** за проверу једнакости два објекта је дефинисан у класи Object. У датом програмском коду у класи **Klasa** је надјачан метод **equals()**. Анализирати код и проценити тачност извршавања.

```
public class Test {
    public static void main(String[] args) {
        Object obj1 = new Klasa();
        Object obj2 = new Klasa();
        System.out.println(obj1.equals(obj2));
    }
}
class Klasa {
    int x;
    public boolean equals(Object o) {
        Klasa a = (Klasa)o;
        return this.x == a.x;
    }
}
```

2

Заокружити број испред очекиваног одговора:

1. Програм има грешку, јер се изразом obj1.equals(obj2) проверава једнакост објекта obj1 и obj2 различитог типа од Object.
2. Програм има грешку, јер се једнакост објекта obj1 и obj2 типа **Klasa** проверава изразом obj1 == obj.
3. Програм се извршава без грешке и приказује се **true** на екрану.
4. Програм се извршава без грешке и приказује се **false** на екрану

293. У програмском језику Јава, метод **equals()** за проверу једнакости два објекта је дефинисан у класи Object. У датом програмском коду, у класи **Klasa** је предефинисан метод **equals()**. Анализирати код и проценити тачност извршавања.

```
public class Test {
    public static void main(String[] args) {
        Object obj1 = new Klasa();
        Object obj2 = new Klasa();
        System.out.println(obj1.equals(obj2));
    }
}
class Klasa {
    int x;
    public boolean equals(Klasa o) { return this.x == a.x; }
}
```

2

Заокружити број испред очекиваног одговора:

1. Програм има грешку, јер се изразом obj1.equals(obj2) проверава једнакост објекта obj1 и obj2 различитог типа од Object.
2. Програм има грешку, јер се једнакост објекта obj1 и obj2 типа **Klasa** проверава изразом obj1 == obj.
3. Програм се извршава без грешке и приказује се **true** на екрану.
4. Програм се извршава без грешке и приказује се **false** на екрану

294. Дат је код програма у програмском језику Јава. У main() методи декларисане су променљиве **s**, **obj** и **t**. Анализирати декларацију и одредити на који објекат указују променљиве **s**, **obj** и **t**.

```
public class TestPrimer {
    public static void main(String[] args) {
        String s = "Java";
        Object obj = s;
        String t = (String) obj;
    }
}
```

2

Заокружити број испред очекиваног одговора:

1. Када се вредност променљиве **s** додељује променљивој **obj** у наредби **Object obj = s**, конструише се нови објекат.
2. Када се конвертује тип променљиве **obj** и њена вредност додељује променљивој **t** у наредби **String t = (String)obj**, конструише се нови објекат.
3. Када се конвертује тип променљиве **obj** и њена вредност додељује променљивој **t** у наредби **String t = (String)obj**, садржај променљиве **obj** се мења.
4. Променљиве **s**, **obj** и **t** указују на исти објекат типа **String**.

295. Дат је код програма у програмском језику Јава који дефинише класу **Osoba** са њеним методама и атрибутима. Анализирати дати код и одредити који од понуђених одговора је потребно дописати у 1. ред кода како би метода била тачно дефинисана.

```
1. public class Osoba _____ {
2.     private String ime;
3.     private String prezime;
4.     int godina;
5.     public Osoba(String ime, String prezime, int godina){
6.         this.ime=ime;
7.         this.prezime=prezime;
8.         this.godina=godina;
9.     }
10.    public int compareTo(Osoba osb){
11.        if(this.godina>osb.getGodina()) return 1;
12.        else if(this.godina<osb.getGodina()) return -1;
13.        else return 0;
14.    }
15.    public String toString(){
16.        return "Ime: "+this.getIme()+"\tPrezime: " +
17.            this.getPrezime() + "\tGodina: " + this.getGodina();
18.    }
}
```

1

Заокружити број испред тачног одговора:

1. implements Comparable<Osoba>
2. extends Comparable
3. implements Comparable
4. interface Comparable<Osoba>

296. Дат је код програма у програмском језику Јава који дефинише класу Osoba са њеним методама и атрибутима. Анализирати дати код и на основу декларације метода CompareTo(...) и Clone() одредити код које недостаје у првој линији.

```

1. public class Osoba implements _____ {
2.     private String ime;
3.     private double dohodak;
4.     private Date datum;
5.     public Osoba(String ime,double dohodak){
6.         this.ime=ime;
7.         this.dohodak=dohodak;
8.     }
9.     public int compareTo(Osoba osb){
10.         if(this.dohodak<osb.dohodak) return -1;
11.         else if(this.dohodak>osb.dohodak) return 1;
12.         else return 0;
13.     }
14.     public Osoba clone() throws CloneNotSupportedException {
15.     {
16.         Osoba cloned = (Osoba)super.clone();
17.         cloned.datum = (Date)datum.clone();
18.         return cloned;
19.     }
20.     public String toString(){
21.         return "Ime: "+this.ime+"\nDohodak: "+this.dohodak+"\nDatum:
22.             "+this.datum;
22.     }
23. }
```

2

Заокружити број испред тачног одговора:

1. Comparable<Osoba>, Cloneable
2. Comparable, Cloneable<Osoba>
3. Comparable<Osoba> и Cloneable<Osoba>
4. Comparable, Cloneable

297. Дат је код програма у програмском језику Јава којим су дефинисане две класе: **class Test** која садржи **main(String[] args)** методу и **class A**. Анализирати дати код и одредити да ли је код исправно написан. Понуђени одговори дају опис последица извршавања овог кода. Заокружити број испред тачног исказа..

```

public class Test {
    public static void main(String[] args) {
        A a1 = new A();
        A a2 = new A();
        System.out.println(a1.equals(a2));
    }
}
class A {
    int x;
    public boolean equals(A a) {
        return this.x == a.x;
    }
}
```

2

1. Програм има грешку, јер се изразом **a1.equals(a2)** проверава једнакост објеката **a1** и **a2** различитог типа од **Object**.
2. Програм има грешку, јер се једнакост објеката **a1** и **a2** типа **A** проверава изразом **a1 == a2**.
3. Програм се извршава без грешке и приказује се **true** на екрану.
4. Програм се извршава без грешке и приказује се **false** на екрану

298. Дат је код програма у програмском језику Јава којим су дефинисане три класе: **class** Test која садржи main(String[] args) методу, **class** Vozilo и **class** Auto. Компајлер приликом превођења овог кода јавља грешку. Заокружити број испред одговора у коме је написано који редови су изазвали грешку приликом компајлирања.

```

1. public class Test {
2.
3.     public static void main(String[] args) {
4.         Vozilo v = new Vozilo(8);
5.         v.vazi();
6.         v.brojVrata = 2;
7.         Vozilo bmw = new Auto(2, 4);
8.         bmw.vazi();
9.         Auto audi = new Auto(4);
10.        Audi.vazi();
11.    }
12. }
13. class Vozilo {
14.     public int brojTočkova;
15.     public Vozilo(int t) {
16.         brojTočkova = t;
17.     }
18.     public void vazi() {
19.         System.out.println("Vožnja vozila");
20.     }
21. }
22. class Auto extends Vozilo {
23.     public int brojVrata;
24.     public Auto(int v, int t) {
25.         super(t);
26.         brojVrata = v;
27.     }
28.     public void vazi() {
29.         System.out.println("Vožnja auta");
30.     }
31. }
```

1. Грешке су у редовима 7, 10 i 16.
2. Грешке су у редовима 7, 9 i 22.
3. Грешке су у редовима 6, 7 i 10.
4. Грешке су у редовима 6, 9 i 10.
5. Грешке су у редовима 6, 9 i 25.
6. Грешке су у редовима 9, 10 i 25.

3

299. Дат је код програма у програмском језику Јава који дефинише интерфејс **Poredjenje**. Интерфејс Poredjenje садржи декларацију методе **porediPovrsine()**, која пореди објекте типа Figura. У датом коду дописати у 3. линији кода наредбу која недостаје да би метода била исправно декларисана.

```

1. package figure;
2.
3. public _____ {
4.     int porediPovrsine(Figura fig);
5. }
```

1

Заокружити број испред одговора који даје исправно решење:

1. extends Poredjenje
2. interface Poredjenje
3. implements Poredjenje
4. abstract Poredjenje

300. Дат је код програма у програмском језику Јава у ком су дефинисане две класе: **class Test** која садржи **main(String[] args)** методу и **class A**. Анализирати дати код и одредити да ли је код исправно написан. Понуђени одговори дају опис последица извршавања овог кода. Заокружити број испред тачног исказа..

```
public class Test {
    public static void main(String[] args) {
        Object a1 = new A();
        Object a2 = new A();
        System.out.println(a1.equals(a2));
    }
}
class A {
    int x;
    public boolean equals(A a) {
        return this.x == a.x;
    }
}
```

2

1. Програм има грешку, јер се изразом `a1.equals(a2)` проверава једнакост објеката `a1` и `a2` различитог типа од `Object`.
2. Програм има грешку, јер се једнакост објеката `a1` и `a2` типа `A` проверава изразом `a1 == a2`.
3. Програм се извршава без грешке и приказује се `true` на екрану.
4. Програм се извршава без грешке и приказује се `false` на екрану.

301. Једна од основних особина објектно оријентисаног језика је наслеђивање. Дате су насловне линије дефиниције класе. Заокружити број испред дефиниције класе која се **НЕ** може наследити.

1. `class A {}`
2. `class A { private A(){ } }`
3. `final class A {}`
4. `class A { protected A(){ } }`

1

302. Дат је део кода који је написан на Јава програмском језику. Анализирати и одредити шта ће се приказати на излазу извршавањем овог кода.

```
class PassA {
    public static void main(String [] args) {
        PassA p = new PassA();
        p.start();
    }
    void start() {
        long [] a1 = {3,4,5};
        long [] a2 = fix(a1);
        System.out.print(a1[0] + a1[1] + a1[2] + " ");
        System.out.println(a2[0] + a2[1] + a2[2]);
    }
    long [] fix(long [] a3) {
        a3[1] = 7;
        return a3;
    }
}
```

3

Заокружи број испред тачног одговора:

1. 12 15
2. 15 15
3. 3 4 5 3 7 5
4. 3 7 5 3 7 5

303. Дат је део кода који је написан на Јава програмском језику. Анализирати код и одредити шта ће бити на излазу.

```

class PassS {
    public static void main(String [] args) {
        PassS p = new PassS();
        p.start();
    }
    void start() {
        String s1 = "slip";
        String s2 = fix(s1);
        System.out.println(s1 + " " + s2);
    }
    String fix(String s1) {
        s1 = s1 + "stream";
        System.out.print(s1 + " ");
        return "stream";
    }
}

```

Заокружи број испред тачног одговора:

1. slip stream
2. slipstream stream
3. stream slip stream
4. slipstream slip stream

2

304. Дате су наредбе које су написане на Јава програмском језику. Анализирати дате наредбе и одредити која од њих је исправно написана. Заокружити број испред тачног одговора:

1. int a = Math.abs(-5);
2. int b = Math.abs(5.0);
3. int c = Math.abs(5.5F);
4. int d = Math.abs(5L);

1

305. Дати су позиви методе Math.max дефинисане у Јава програмском језику. Анализирати дате позиве и одредити који од њих су исправно написани. Заокружити број испред **НЕИСПРАВНО** написаног позива методе:

1. Math.max(1, 4);
2. Math.max(2.3, 5);
3. Math.max(1, 3, 5, 7);
4. Math.max(-1.5, -2.8f);

1

306. Заокружити број испред исказа који представља исправан наставак дате тврђње:

При креирању објекта изведене класе...

1. извршава се само конструктор изведене класе
2. прво се извршава конструктор родитељске класе, али само ако је позван кључном речју **super**
3. обавезно се прво извршава конструктор изведене, а потом конструктор родитељске класе
4. обавезно се прво извршава конструктор родитељске, а потом конструктор изведене класе

1

307. Дат је код програма у програмском језику Јава који дефинише апстрактну класу Figura. У класи Figura написати статички иницијални блок у ком се иницијализује вредност атрибута *brojFigura*. У 10. линију кода додати наредбу која дефинише статички иницијални блок.

```

1. package figure;
2. public abstract class Figura implements Poredjenje{
3.     protected static int brojFigura;
4.     protected String naziv;
5.     public Figura(String naziv){
6.         this.naziv=naziv;
7.     }
8.     public abstract double povrsinaFigure();
9.     public abstract double obimFigure();
10.    public static void uvecajBrojFigura(){
11.        brojFigura++;
12.    }
13.    public String toString(){
14.        return "Figura ";
15.    }
16. }
17. }
```

2

Заокружити број испред одговора који даје исправно решење:

1. { *brojFigura*=0; }
2. *brojFigura*=0;
3. static { *brojFigura*=0; }
4. static Figura { *brojFigura*=0; }

308. Дат је код програма у програмском језику Јава у ком су дефинисане три класе: **class** Test која садржи main(String[] args) методу, **class** A и **class** B. Анализирати дати код и одредити шта ће се приказати на екрану као резултат извршавања овог програма. Заокружити број испред одговора који садржи резултат исписа.

```

public class Test extends A {
    public static void main(String[] args) {
        Test t = new Test();
    }
}
class A extends B {
    public A() { System.out.println("Pozvan podrazumevani konstruktor klase A"); }
}
class B {
    public B() { System.out.println("Pozvan podrazumevani konstruktor klase B"); }
}
```

2

1. Ништа.
2. Позван подразумевани конструктор класе A
3. Позван подразумевани конструктор класе B
4. Позван подразумевани конструктор класе A и у другом реду Позван подразумевани конструктор класе B
5. Позван подразумевани конструктор класе B и у другом реду Позван подразумевани конструктор класе A

309. У програмском језику Јава користи се службена реч ***super***. Проценити који од наредних исказа који дефинишу дату службену реч **НИЈЕ** тачан.

Заокружити број испред очекиваног одговора:

1. Службена реч ***super*** може послужити за позивање конструктора родитељске класе.
2. Службена реч ***super*** може послужити за позивање приватних метода родитељске класе којима се другачије не може приступити.
3. Службена реч ***super*** може послужити за позивање заклоњеног метода родитељске класе.
4. Службена реч ***super*** може послужити за позивање заклоњеног поља родитељске класе.

1

310. Дат је код програма у програмском језику Јава у ком су дефинисане три класе: ***class Test*** која садржи ***main(String[] args)*** методу, ***class A*** и ***class B***. Анализирати дати код и одредити да ли је код исправно написан. Заокружити број испред исказа који даје информацију о тачности кода.

```
public class Test {
    public static void main(String[] args) {
        B b = new B();
        b.metod(5);
        System.out.println("b.i je " + b.i);
    }
}
class A {
    int i;
    public void metod(int i) { this.i = i; }
}
class B extends A {
    public void metod(String s) { System.out.println(s); }
}
```

2

1. Програм има грешку, јер је метод ***metod(int i)*** надјачан са различитим потписом у класи **B**.
2. Програм има грешку, јер се ***b.metod(5)*** не може позвати пошто је метод ***metod(int i)*** заклоњен у класи **B**.
3. Програм има грешку због ***b.i***, јер је поље *i* неприступачно из класе **B**.
4. Програм нема грешке, јер наслеђени метод класе **A**, ***metod(int i)*** није надјачан у класи **B**, већ је дефинисан преоптерећен метод ***metod(String s)***.

311. Заокружити број испред исказа који представља исправан наставак дате реченице:

Ако try-catch наредба има више catch блокова у којима „хватамо“ изузетак основне ***Exception*** класе, заједно са изузетцима других класа изведенних из класе ***Exceptions...***

2

1. онда се изузетак основне *Exception* класе може „хватати“ у било ком catch блоку (редослед није битан, битно је да се наведу све могуће грешке)
2. онда се изузетак основне *Exception* класе мора „хватати“ у последњем catch блоку
3. онда се изузетак основне *Exception* класе мора „хватати“ у првом catch блоку
4. основна *Exception* класа се не комбинује у истој наредби са класама изведеним из ње јер их осавна класа „маскира“

312. Дат је део кода који је написан на Јава програмском језику. Анализирати код и одредити шта ће се приказати на излазу.

```
try
{
    int x = 0;
    int y = 5 / x;
}
catch (Exception e)
{
    System.out.println("Exception");
}
catch (ArithmetricException ae)
{
    System.out.println(" Arithmetic Exception");
}
System.out.println("finished");
```

2

Заокружи број испред тачног одговора:

1. Приказује се текст: finished
2. Приказује се текст: Exception
3. Ништа. Дешава се грешка приликом компајлирања
4. Приказује се текст: Arithmetic Exception

313. Дат је део кода који је написан на Јава програмском језику. Анализирати код и одредити шта ће бити на излазу.

```
public class RTEExcept {
    public static void throwit () {
        System.out.print("throwit ");
        throw new RuntimeException();
    }
    public static void main(String [] args) {
        try {
            System.out.print("hello ");
            throwit();
        }
        catch (Exception re ) { System.out.print("caught "); }
        finally { System.out.print("finally "); }
        System.out.println("after ");
    }
}
```

2

Заокружи број испред тачног одговора:

1. hello throwit caught
2. Грешка приликом компајлирања
3. hello throwit *RuntimeException* caught after
4. hello throwit caught finally after

У следећим задацима заокружите бројеве испред тражених одговора

314. Дати су искази који у програмском језику Јава дефинишу конструктор.
Заокружити бројеве испред очекиваних одговора:
1. Подразумевани конструктор без параметара се увек аутоматски додаје класи.
 2. Подразумевани конструктор без параметара се класи аутоматски додаје уколико у њој није експлицитно дефинисан ниједан конструктор.
 3. У класи се мора експлицитно дефинисати бар један конструктор.
 4. Конструктори немају тип резултата, чак ни **void**.
315. Дати код програма у програмском језику Јава састоји се од две класе у једној датотеци.
Анализирати дати код и проценити његову тачност.
Заокружити бројеве испред тачних исказа:
- ```

public class proba {
 public static void main(String[] args) {
 A a = new A();
 a.print();
 }
}
class A {
 String s;
 public A(String s) { this.s = s; }
 public void print() {
 System.out.println(s);
 }
}
```
1. Програм има грешку, јер класа А није јавна класа.
  2. Програм има грешку, јер класа А нема подразумевани конструктор.
  3. Програм нема грешака и нормално се извршава ништа не приказујући на екрану.
  4. Програм има грешку која се може исправити уколико се у трећем реду наредба **A a = new A();** замени наредбом **A a = new A("poruka");**
316. Дата је дефиниција класе у програмском језику Јава и састоји се од два конструктора и поља x и y. У четвртом реду допунити подразумевани конструктор без параметара класе Point који формира тачку у координатном почетку, позивајући конструктор са параметрима.
1. **public class** Point {
  2.     **private double** x,y;
  3.     **public** Point(**double** x, **double** y) { **this**.x=x; **this**.y=y; }
  4.     **public** Point() { \_\_\_\_\_; }
  5. }
- Заокружити бројеве испред тачних одговора:
1. super(0,0)
  2. this(0,0)
  3. Point(0,0)
  4. this(0.0,0.0)
  5. Point(0.0,0.0)

317. Дата је дефиниција класе у програмском језику Јава и састоји се од два конструктора, једне методе и поља x. У дефиницији се користи службена реч **this**. Анализирати дати код и проценити тачност следећих исказа. Заокружити бројеве испред тачних исказа:

```
public class TestPrimer {
 private double x;
 public TestPrimer(double x) {
 this.fun();
 this.x = x;
 }
 public TestPrimer() {
 System.out.println("Podrazumevani konstruktor");
 this(23);
 }
 public void fun() {
 System.out.println("Poziv metoda fun()");
 }
}
```

1. **this.fun()** у конструктору `TestPrimer(double x)` може се поједноставити и заменити само са `fun()`.
2. **this.x** у конструктору `TestPrimer(double x)` може се поједноставити и заменити само са `x`.
3. **this(23)** у конструктору `TestPrimer()` мора се писати пре наредбе `System.out.println("Podrazumevani konstruktor");`.
4. **this(23)** у конструктору `TestPrimer()` мора се заменити са прецизнијим изразом `this(23.0)`.

2

2

2

318. Дати су искази који дефинишу иницијализацију почетних вредности променљивих у програмском језику Јава. Заокружити бројеве испред **НЕТАЧНИХ** исказа:

1. Локалне променљиве метода добијају аутоматски почетне вредности.
2. Глобалне променљиве (поља) класе добијају аутоматски почетне вредности.
3. Променљива неког примитивног типа садржи вредност тог примитивног типа.
4. Променљива неког класног типа садржи меморијску адресу на којој се налази објекат тог класног типа.
5. Променљивој класног типа може се доделити цео број који представља важећу меморијску адресу.
6. При додели примитивне промељиве (**a**), другој примитивној променљивој (**b**), врши се копирање вредности промељиве **a** у променљиву **b**, чиме се добијају две независне копије исте вредности

319. Дата је дефиниција класе у програмском језику Јава и састоји се од два конструктора, методе и поља x и y. У петом реду дефинисан је конструктор са параметрима који формира тачку са координатама x и y. Заокружити наредбе којима се може допунити дефиниција конструктора:

```
1. public class Point {
2. private double x,y;
3. public Point() { x=0; y=0; }
4. public void set(double xx,double yy) { x=xx; y=yy; }
5. public Point(double x,double y) { _____; }
```

Заокружити бројеве испред тачних одговора:

1. `this.x=x; this.y=y;`
2. `x=x; y=y;`
3. `set(x,y);`
4. `set(this.x,this.y);`
5. `x=this.x; y=this.y;`

320. Дата је наредба кода у програмском језику Јава која представља декларацију низа. Проценити које од доле наведених декларација су тачне.  
Заокружити бројеве испред очекиваних одговора:

1. int niz = new int(30);
2. double[ ] niz = new double[30];
3. int[ ] niz = { 3, 4, 3, 2 };
4. char[ ] niz = new char[ ];
5. char[ ] niz = new char { 'a', 'b', 'c', 'd' };
6. char[ ] niz = new char[ ] { 'a', 'b' };

1,5

321. Дати су делови кода у програмском језику Јава који треба да рачунају збир елемената матрице **a** типа **int[][]**. Анализирати дате кодове и проценити који од предлога је тачан. Заокружити бројеве испред очекиваних одговора:

1. **int** s=0;  
**for**(**int** i=0; i<a.length; i++)  
    **for**(**int** j=0; j<a.length; j++) s+=a[i][j];
2. **int** s=0;  
**for**(**int**[ ] vrsta : a)  
    **for**(**int** el : vrsta) s+=el;
3. **int** s=0;  
**for**(**int** i=0; i<a.length; i++)  
    **for**(**int** j=0; j<a[i].length; j++) s+=a[i][j];
4. **int** s=0;  
**for**(**int**[ ] vrsta : a)  
    **for**(**int**[ ] el : vrsta) s+=el;

2

322. Дати искази дефинишу особине наслеђивања. Заокружити бројеве испред тачних исказа:

1. Једна класа у Јави може директно бити наслеђена из више класа.
2. Наслеђена класа садржи додатна поља и методе у односу на своју родитељску класу.
3. „Класа А наслеђује класу Б” значи да је А поткласа класе Б.
4. Ако класа А наслеђује класу Б, тада објекти класе А садрже сва поља и све методе класе Б.
5. Ако класа А наслеђује класу Б, тада се за сваки објекат класе А подразумева да је истовремено и објекат класе Б.
6. У Јави је могуће из једне у другу класу наследити само методе и својства, али не и поља класе.

1,5

323. Да би наслеђени метод могао да се **редефинише** и тиме измени његова функционалност у класама наследницима, у родитељској класи испред ознаке повратног типа метода **НЕ СМЕ** се навести нека од понуђених кључних речи.

Заокружити бројеве испред кључних речи на које се односи претходна тврдња:

1. final
2. override
3. abstract
4. private
5. default, када је класа наследница у другом пакету
6. super
7. public или protected

1,5

324. На програмском језику Јава дефинисане су класе:

```

1. class A {
2. public int Metod() { return 10; }
3. }
4. class B extends A {
5. public final int Metod() { return 20; }
6. }
7. class C extends B {
8. public int Metod() { return 30; }
9. }

```

2

Компилатор јавља грешку при превођењу овог кода коју је могуће решити на више начина у зависности од очекиваног ефекта.

Заокружити бројеве испред понуђених решења која ће по вашем мишљењу отклонити грешку у коду:

1. У 5. линији кода метод у класи В прогласити за **abstract** уместо **final**
2. У 5. линији кода избрисати кључну реч **final**
3. У 7. линији кода класу С наследити из класе А, уместо из класе В
4. У 8. линији кода, иза дефиниције методе у класи С, позвати основни метод :**super()**
5. У 8. линији кода метод у класи С такође прогласити за **final**

325. У класи Figura дат је подразумевани (default) конструктор и конструктор са 4 параметра:

```

public Figura() {...}
public Figura(String ime, String boja, int pozX, int pozY) {...}

```

Заокружити бројеве испред исправно написаних наредби креирања објекта класе Figura:

1. Figura f = Figura("lovac", "beli", 7, 3);
2. Figura f = new Figura("beli", "lovac", 7, 3);
3. Figura f = new Figura();
4. Figura f = new Figura("lovac", 7, 3, "beli");
5. Figura f = new Figura("lovac", "beli", 7, 3);
6. Figura f = new Figura("lovac", "beli", 3);

3

326. Заокружити бројеве испред наведених чланова класе који се ни под којим условима **НЕ** наслеђују са родитељске класе на изведену класу:

1. Final поља
2. Заштићени чланови класа
3. Сетери и гетери
4. Приватни чланови класа
5. Конструктор класе

2

327. Дати су искази који програмском језику Јава дефинишу рекурзивне методе.

Заокружити бројеве испред **НЕТАЧНИХ** исказа:

1. Рекурзивни методи су они који позивају сами себе, било директно или индиректно.
2. Рекурзивни методи се позивају другачије од нерекурзивних метода.
3. Применом рекурзије у решавању проблема увек долази до побољшања перформанси програма
4. Рекурзивни методи решавају неки задатак својењем полазног проблема на сличан, простији проблем (или више њих).
5. Сваки рекурзивни метод мора имати базни случај за најпростији задатак чије се решење не добија рекурзивним позивом.

2

328. Дате су наредбе у програмском језику Јава које дефинишу заглавље методе print() са променљивим бројем параметара. Одредити који од понуђених одговора су исправни.

Заокружити бројеве испред очекиваних одговора:

1. `public void print(String... niska, double... broj)`
2. `public void print(double... broj, String ime)`
3. `public void double... print(double d1, double d2)`
4. `public void print(double... broj)`
5. `public void print(int n, double... broj)`

2

329. Дати су искази који дефинишу полиморфизам, једну од особина објектно-оријентисаног језика Јава. Анализирати исказе и заокружити бројеве испред тачних исказа:

1. Када је класа Б изведена из класе А, објекту класе А може се доделити показивач на објекат класе Б
2. Када је класа Б изведена из класе А, у генеричку листу објеката класе Б, могу се убацити и објекти класе А и објекти класе Б
3. Објекат типа Б се може пренети као аргумент методу на место параметра типа А уколико је Б класа наследница од А.
4. Када се у класи Б која је изведена из класе А изврши предефинисање (override) наслеђене методе, онда су у објекту класе Б доступне обе верзије ове методе (исто као код преоптеређивања метода – overload)

2

330. Дат је код програма у програмском језику Јава. Код садржи објекте две класе у којима је дефинисан метод `toString()`. Анализирати код датог програма и одредити који од датих исказа су тачни.

```
public class TestPrimer {
 public static void main(String[] args) {
 Object a = new Klasa();
 Object obj = new Object();
 System.out.println(a);
 System.out.println(obj);
 }
}
class Klasa {
 int x;
 public String toString() { return "x у А је " + x; }
}
```

2

Заокружити бројеве испред очекиваних одговора:

1. Програм има грешку, јер наредбу `System.out.println(a)` треба заменити наредбом `System.out.println(a.toString())`.
2. Приликом извршавања наредбе `System.out.println(a)`, програм позива се метод `toString()` наслеђен из класе `Object`.
3. Приликом извршавања наредбе `System.out.println(a)`, програм позива метод `toString()` из класе А.
4. Приликом извршавања наредбе `System.out.println(obj)`, програм позива метод `toString()` из класе `Object`.

331. Дат је код програма у програмском језику Јава у ком су дефинисане четири класе: **class Test** која садржи **main(String[] args)** методу, **class MasterStudent**, **class Student** и **class Osoba**. Приликом превођења овог кода компајлер јавља грешку. Заокружити бројеве испред понуђених исказа који описују разлоге настајања грешке приликом компајлирања.

```
public class Test {
 public static void main(String[] args) {
 m(new MasterStudent());
 m(new Student());
 m(new Osoba());
 m(new Object());
 }
 public static void m(Student x) {
 System.out.println(x.toString());
 }
}
class MasterStudent extends Student {}
class Student extends Osoba {
 public String toString() { return "Student"; }
}
class Osoba extends Object {
 public String toString() { return "Osoba"; }
}
```

2

1. Грешка је у позиву **m(new MasterStudent())**.
2. Грешка је у позиву **m(new Student())**.
3. Грешка је у позиву **m(new Osoba())**.
4. Грешка је у позиву **m(new Object())**.

332. У програмском језику Јава дата је декларација класе **A** и класе **TestPrimer** која наслеђује класу A. Анализирати дате класе и проценити који од понуђених исказа су тачни. Заокружити бројеве испред тачних исказа:

```
public class TestPrimer extends A {
 public static void main(String[] args) {
 TestPrimer t = new TestPrimer();
 t.print();
 }
}
class A {
 String s;
 public A(String s) { this.s = s; }
 public void print() { System.out.println(s); }
}
```

2

1. Програм има грешку, јер класа TestPrimer нема подразумевани конструктор TestPrimer () .
2. Програм има грешку, јер класа TestPrimer има имплицитни подразумевани конструктор без параметара TestPrimer (), али наслеђена класа A нема такав конструктор. Програм би радио без грешке уколико би се уклонио конструктор у класи A.
3. Програм има грешку, али би радио без грешке уколико би се класи A експлицитно додао конструктор без параметара A().
4. Програм нема грешку, извршава се, али се на конзоли ништа не исписује јер је поље **s** објекта **t** добило подразумевану вредност празног стринга

333. Дат је код написан у Јави којим су креиране три класе у ланцу наслеђивања. Имајући у виду класификаторе приступа пољима класа, заокружити бројеве испред поља која ће бити видљива унутар класе Sin:

```
public class Deda {
 private double penzija;
 protected String adresa;
 private int godinaRodjenja;
 public String ime;
}
public class Otac extends Deda {
 private double plata;
 protected String struka;
 private string brojRacuna;
}
public class Sin extends Otac {
 public int razred;
}
```

1. penzija
2. adresa
3. ime
4. plata
5. struka
6. razred
7. godinaRodjenja
8. brojRacuna

2

334. Дат је код на Јави којим су креиране три класе у ланцу наслеђивања. Имајући у виду класификаторе приступа пољима класа, заокружити бројеве испред поља која ће бити видљива унутар класе Sin:

```
public class Deda {
 private double penzija;
 protected String adresa;
 public String ime;
}
public class Otac extends Deda {
 private double plata;
 protected String struka;
}
public class Sin extends Deda {
 public int razred;
}
```

1. penzija
2. adresa
3. ime
4. plata
5. struka
6. razred

1,5

335. Дат је код написан у Јави којим су креиране три класе у ланцу наслеђивања. Унутар сваке класе декларисан је по један private, public и protected атрибут. У методи **Main()** класе **Program** креiran је објекат s класе **Sin** (**Sin** s = new **Sin**();) Заокружити бројеве испред поља која ће бити видљива у креiranом објекту s класе Sin:

```
public class Deda {
 private double penzija;
 protected String adresa;
 public String ime;
}
public class Otac extends Deda {
 private double plata;
 protected String firma;
 public String struka;
}
public class Sin extends Otac {
 private double prosek;
 protected int razred;
 public String skola;
}
```

1. penzija
2. adresa
3. ime
4. plata
5. struka
6. firma
7. prosek
8. razred
9. skola

1,5

336. Дат је код програма у програмском језику Јава који декларише два објекта типа **Pravougaonik** и један објекат типа **Krug**. Класе Pravougaonik и Krug наслеђују апстрактну класу **Figura** и обе класе имају методу **toString**. Код програма треба допунити са наредбом која ће исписивати низ **Figura[]** декларисан у коду програма.

```

1. public abstract class Figura {
2. protected static int brojFigura=0;
3. public static int GetBrojFigura() { return brojFigura; }
4. public Figura(){ brojFigura++; }
5. public abstract double povrsinaFigure();
6. public String toString(){ return "Figura "; }
7. }
...
1. Pravougaonik pr1 = new Pravougaonik();
2. Pravougaonik pr2 = new Pravougaonik(3,2);
3. Krug kr1 = new Krug(2);
4.
5. System.out.println("Broj figura="+Figura.GetBrojFigura()+"\n");
6. Figura[] niz = new Figura[Figura.GetBrojFigura()];
7. niz[0]=pr1; niz[1]=pr2; niz[2]=kr1;
8.
9. System.out.println("Prikaz figura:");
10.

```

2

Заокружити бројеве испред одговора који представљају могуће начине приказа низа фигура:

1. **for(Figura el:niz) System.out.println(el);**
2. **for(int i=0; i<niz.length; i++) System.out.println(niz[i]);**
3. **for (el:niz) System.out.println(el);**
4. **for(int i:niz) System.out.println(niz[i]);**

337. Дати су изкази који се односе на правила писања try-catch-finally блокова за руковање изузетима. Заокружити бројеве испред исказа који су тачни:

1. Блок try мора имати бар један catch блок
2. Блок try може имати више catch блокова
3. Ако блок try има више catch блокова, изузетак основне Exception класе мора се хватати у првом catch блоку
4. Ако блок try има више catch блокова, битан је редослед њиховог писања
5. Блок try мора имати бар један finally блок
6. Блок try не сме да има више catch блокова

1,5

### Допуните следеће реченице и табеле

338. Допунити дате изјаве уносом кључне речи која недостаје:

Промена вредности приватног поља врши се помоћу метода које се називају \_\_\_\_\_ док се читање вредности приватног поља врши помоћу метода које се називају \_\_\_\_\_.

1

339. Са леве стране, у првој колони су наведени елементи, а у другој класе класификатори приступа елементима. Допуните реченицу која се односи на подразумевано право приступа наведеним елементима.

На линију у уписати одговарајуће појмове наведене у колонама:

|                     |              |                                                |
|---------------------|--------------|------------------------------------------------|
| 1. Класа            | 1. private   | Сви елементи наведени у првој колони           |
| 2. Полье класе      | 2. protected | са лева имају подразумевани                    |
| 3. Својство у класи | 3. public    | класификатор приступа _____,                   |
| 4. Метод класе      | 4. default   | осим _____ где је подразумевани приступ _____. |
| 5. Конструктор      |              |                                                |

2

340. Допунити дате изјаве уносом кључне речи која недостаје:

Метод дефинисан у родитељској класи, у класама наследницама може бити редефинисан или скривен.

1

**Сакривање** методе родитељске класе врши се тако што у изведеном класи постоји метода истог \_\_\_\_\_.

341. На програмском језику Јава дефинисане су две класе:

```
public class Racun {
 public int Uvecaj() { return 10; }
}
public class Dinarski extends Racun {
 public int Uvecaj() { return 20 * super.Uvecaj(); }
}
public class Devizni extends Dinarski {
 public int Uvecaj() { return 50 + super.Uvecaj(); }
}
```

Унутар функције Main, креирана су три објекта ових класа на следећи начин:

3

```
Racun r = new Racun();
Racun rDin = new Dinarski();
Racun rDev = new Devizni();
```

Анализирати код и на предвиђене линије уписати шта метод Uvecaj() враћа при позиву из наведених објеката:

r.Uvecaj(); \_\_\_\_\_  
rDin.Uvecaj(); \_\_\_\_\_  
rDev.Uvecaj(); \_\_\_\_\_

342. На програмском језику Јава дефинисане су две класе:

```
public class Racun {
 public int Uvecaj() { return 10; }
}
public class Dinarski extends Racun {
 public int Uvecaj() { return 20 * super.Uvecaj(); }
}
public class Devizni extends Racun {
 public int Uvecaj() { return 50 + super.Uvecaj(); }
}
```

Унутар функције Main, креирана су три објекта ових класа на следећи начин:

3

```
Racun r = new Racun();
Racun rDin = new Dinarski();
Racun rDev = new Devizni();
```

Анализирати код и на предвиђене линије уписати шта метод Uvecaj() враћа при позиву из наведених објеката:

r.Uvecaj(); \_\_\_\_\_  
rDin.Uvecaj(); \_\_\_\_\_  
rDev.Uvecaj(); \_\_\_\_\_

343. На програмском језику Јава дефинисане су две класе:

```
class Roditelj{
 public void Poruka1(){ System.out.println("R1"); }
 public static void Poruka2(){ System.out.println("R2"); }
}
class Dete extends Roditelj{
 public void Poruka1(){ System.out.println("D1"); }
 public static void Poruka2(){ System.out.println("D2"); }
}
```

Унутар функције main, креирана су два објекта ових класа на следећи начин:

4

```
Dete x = new Dete();
Roditelj y = new Dete();
```

Проценити ефекат извршења наведених позива и на предвиђене линије уписати шта ће се видети на стандардном излазу извршењем позваних метода:

x.Poruka1(); \_\_\_\_\_  
x.Poruka2(); \_\_\_\_\_  
y.Poruka1(); \_\_\_\_\_  
y.Poruka2(); \_\_\_\_\_

344. Дате су започете изјаве које се односе на делове кода за обраду изузетака. Довршити започете реченице:

Наредбе које се извршавају у случају настанка грешке, стављају се унутар блока \_\_\_\_\_

1,5

Наредбе које се извршавају и ако се деси и ако се не деси грешка, стављају се унутар блока \_\_\_\_\_

Наредбе које могу уззвати грешку стављају се унутар блока \_\_\_\_\_

345. На програмском језику Јава дефинисане су две класе:

```
class KlasaA {
 public static int Metod() { return 10; } }
class KlasaB extends KlasaA {
 public static int Metod() { return 20; } }
class KlasaC extends KlasaB {
 public static int Metod() { return 30; } }
```

Креирани су објекти ових класа и из њих позвана метода **Metod()**. На предвиђене линије уписати шта метод **Metod()** враћа при позиву из наведених објеката:

KlasaA a = new KlasaA(); a.Metod() враћа вредност \_\_\_\_\_

KlasaB b = new KlasaB(); b.Metod() враћа вредност \_\_\_\_\_

KlasaA bb = new KlasaB(); bb.Metod() враћа вредност \_\_\_\_\_

KlasaC c = new KlasaC(); c.Metod() враћа вредност \_\_\_\_\_

KlasaB cc = new KlasaC(); cc.Metod() враћа вредност \_\_\_\_\_

KlasaA ccc = new KlasaC(); ccc.Metod() враћа вредност \_\_\_\_\_

3

346. На програмском језику Јава дефинисана класа **CVreme**, а затим креиране две променљиве типа **CVreme**:

```
public class CVreme {
 public int sat;
 public int min;
 public CVreme(int h,int m) {
 sat = h; min = m;
 }
}
CVreme cv1 = new CVreme(12,15);
CVreme cv2 = cv1;
```

2

По извршењу наредби доделе вредности пољима променљиве **cv1**, одредити вредности које ће имати одговарајућа поља променљиве **cv2** и уписати их на предвиђену линију (дате наредбе посматрати као секвенцу):

cv1.sat = 24; cv2.sat = \_\_\_\_\_;

cv1 = new CVreme(-10,-20); cv2.sat = \_\_\_\_\_;

### У следећим задацима уредите и повежите појмове према захтеву

347. Са леве стране су наведени делови/елементи класе, а са десне стране улоге поједињих класних елемената. На линију испред описа улоге унети редни број под којим је наведен одговарајући елеменат класе:

- |                           |       |                                    |
|---------------------------|-------|------------------------------------|
| 1. атрибут                | _____ | Опис функционалности објекта класе |
| 2. garbage collector      | _____ | Контрола приступа пољима класе     |
| 3. конструктор            | _____ | Опис особина објекта класе         |
| 4. метод                  | _____ | Креирање објекта класе             |
| 5. методе сетери и гетери | _____ | Уништавање објекта класе           |

2,5

|      |                                                                                                                                                                                                                                                                           |                                                                                                                        |     |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|-----|
| 348. | Са десне стране наведене су области видљивости појединих елемената класе, а са леве стране модификатори приступа којима се врши контрола области видљивости. На линију испред области видљивости унети редни број под којим је наведен одговарајући модификатор приступа: |                                                                                                                        |     |
| 1.   | private                                                                                                                                                                                                                                                                   | видљив унутар класе у којој је дефинисан, као и унутар изведенih класа                                                 | 2   |
| 2.   | public                                                                                                                                                                                                                                                                    | видљив само унутар класе у којој је дефинисан                                                                          |     |
| 3.   | protected                                                                                                                                                                                                                                                                 | видљив унутар пакета у коме је дефинисан                                                                               |     |
| 4.   | default                                                                                                                                                                                                                                                                   | видљив изван пакета у ком је дефинисан                                                                                 |     |
| 349. | Са леве стране дате су врсте променљивих у програмском језику Јава, а са десне декларације променљивих. На линију испред декларације променљиве унети број којим је означена одговарајућа врста променљиве:                                                               |                                                                                                                        |     |
| 4.   | Променљива вредносног типа                                                                                                                                                                                                                                                | <code>const double eps0 = 8.85E-12</code>                                                                              | 2,5 |
| 5.   | Променљива референтног типа                                                                                                                                                                                                                                               | <code>int x = 10;</code>                                                                                               |     |
| 6.   | Именована константа                                                                                                                                                                                                                                                       | <code>int[] a = new int[7];</code><br><code>int[,] b=new int[10,10];</code><br><code>int[] c = {1,2,3,4,5,6,7};</code> |     |
| 350. | Са леве стране дате су кључне речи које одређују типове класа, а са десне су описи класа. На линију испред описа уписати редни број под којим је наведен одговарајући тип класе:                                                                                          |                                                                                                                        |     |
| 1.   | static                                                                                                                                                                                                                                                                    | Класа из које се не може наслеђивати                                                                                   |     |
| 2.   | abstract                                                                                                                                                                                                                                                                  | Класа садржи само декларације метода, али не и дефиницију (тело) методе                                                | 2   |
| 3.   | final                                                                                                                                                                                                                                                                     | Мора бити креирана као унутрашња (inner) класа                                                                         |     |
| 4.   | interface                                                                                                                                                                                                                                                                 | Класа која се не може инстанцирати                                                                                     |     |
| 351. | Са леве стране наведене су врсте грешака које се могу десити у коду, а са десне описи тих грешака. На линију испред описа уписати редни број под којим је наведен одговарајући тип грешке:                                                                                |                                                                                                                        |     |
| 1.   | Синтаксне грешке<br>(syntax error)                                                                                                                                                                                                                                        | Откривају се у фази компилације и најлакше су за кориговање                                                            | 1,5 |
| 2.   | Логичке грешке<br>(logical error)                                                                                                                                                                                                                                         | Не морају, али се могу десити током извршења и тада доводе до „пуцања“ програма                                        |     |
| 3.   | Грешке у време извршења<br>(run-time error)                                                                                                                                                                                                                               | Програм функционише, нема „пуцања“, али се добијају погрешни резултати                                                 |     |

## ВЕБ ДИЗАЈН

### У следећим задацима заокружите број испред траженог одговора

352. Инернет као глобална мрежа светског система умрежених рачунарских мрежа настао је:
1. Интернет је настало деведесетих година након појаве www интернет сервиса
  2. Почеци интернета се вежу за стварање АРПАНЕТ-а, 1969. године, мреже рачунара под контролом Министарства одбране САД.
  3. Настанак интернета се везује за појаву прве бежичне локалне рачунарске мреже
  4. План интернета је настало деведесетих година али још није у потпуности реализован
353. Која од понуђених изјава описује појам веб сајта:
1. скуп протокола за отпремање и преузимање података са интернета као и протоколи за комуникацију на интернету
  2. скуп веб-страница које могу да садрже текст, слике, видео-снимке и други мултимедијалан садржај састављен у једну целину
  3. веб сајт чини интернет прегледач - програм који се користи за читање хипертекст докумената
  4. интернет сервис (www) која омогућава корисницима да размењују документа која се сastoјe од текста, слика и мултимедијалних компоненти
354. Основна улога веб сервера је:
1. Омогућава израду динамичких веб страница
  2. Веб сервер је систем за управљање релационим базама података на интернету
  3. Веб сервер испоручује захтеване Веб стране корисницима, који су укуцали одређену URL адресу у Веб претраживач
  4. Управља хардверским и софтверским ресурсима рачунара
355. Основни протокол који користи веб сервер је:
1. SMTP
  2. HTTP
  3. HTML
  4. WWW
356. Заокружити број испред линије кода којом се укључује екстерна CSS датотека style.css у оквиру једне HTML странице са циљем дефинисања изгледа елемената дате странице
1. <style type="text/css">....</style>
  2. <link rel="stylesheet" type="text/css" href="style.css"/>
  3. <body style="background-color:red;">...</body>
  4. <a href=" style.css ">CSS датотека</a>
357. Која је основна улога CSS-а (Cascading Style Sheet):
1. Дефинише структуру и садржај странице
  2. Дефинише конкретан изглед елемената странице
  3. Дефинише којим језиком веб сервер извршава динамичку веб страницу
  4. Дефинише протокол за приступ одређеној веб страници

|      |                                                                                                                                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |   |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 358. | Који је од наведених начина за укључивање CSS стила најпогоднији за стилизовање HTML странице, а посебно за касније одржавање и ажурирање:                                                              | 1. Увежена екстерна CSS датотека преко хипервезе у заглављу HTML веб странице - External Style Sheet<br>2. Дефинисан интерни CSS стил у оквиру head секције <code>&lt;style type="text/css"&gt;...&lt;/style&gt;</code> - Internal Style Sheet<br>3. Дефинисан стил у оквиру елемента HTML стране - Inline style<br>4. Не постоји решење за укључивање CSS стила у стилизовање HTML странице                                                                                                                | 2 |
| 359. | Унутар HTML странице, стилови се дефинишу навођењем селектора, својства и вредности. Како се дефинишу селектори стилова:                                                                                | 1. Селектори се дефинишу искључиво на основу имена HTML елемента<br>2. Селектори се дефинишу за HTML елементе на основу имена елемента, назива класе или идентификатора датог елемента<br>3. Селектори се дефинишу на основу атрибута сваког HTML елемента<br>4. Селектори се дефинишу коришћењем кључне речи <code>selector</code> и атрибута                                                                                                                                                              | 1 |
| 360. | Дефинисан је ХТМЛ елемент:                                                                                                                                                                              | <pre>&lt;div id="container"&gt; Maturski ispit - EIT&lt;/div&gt;</pre> <p>Којим начином стила се дефинишу стилови за дати елемент:</p> <ol style="list-style-type: none"> <li>1. container {<br/>              font-size: 1.5em;<br/>              }<br/>2. #container {<br/>              font-size: 1.5em;<br/>              }<br/>3. .container {<br/>              font-size: 1.5em;<br/>              }<br/>4. selector container {<br/>              font-size: 1.5em;<br/>              }</li> </ol> | 2 |
| 361. | Уколико је стил једног DIV елемента дефинисан са три класе <code>blue</code> , <code>black</code> и <code>top</code> , заокружити линију кода којом је елемент исправно дефинисан у оквиру HTML стране: | <ol style="list-style-type: none"> <li>1. &lt;div id="blue black top"&gt;Maturski ispit -EIT&lt;/div&gt;</li> <li>2. &lt;div id="blue" class="black" class="top"&gt;Maturski ispit -EIT&lt;/div&gt;</li> <li>3. &lt;div class="blue" class="black" class="top"&gt;Maturski ispit - EIT&lt;/div&gt;</li> <li>4. &lt;div class="blue black top"&gt;Maturski ispit -EIT&lt;/div&gt;</li> </ol>                                                                                                                 | 2 |
| 362. | Дефинисан је стил елемента у оквиру које се налази позадинска слика. Коју вредност својство <code>background-repeat</code> треба да има уколико слика не треба да се понавља у оквиру елемента:         | <ol style="list-style-type: none"> <li>1. <code>background-repeat: fixed;</code></li> <li>2. <code>background-repeat: null;</code></li> <li>3. <code>background-repeat: no-repeat;</code></li> <li>4. <code>background-repeat: repeat-x;</code></li> </ol>                                                                                                                                                                                                                                                  | 1 |

363. Анализирајте својство дато ниже и заокружите које од понуђених опција одговара својству:

background-attachment: fixed;

дефинисања стила елемента са позадинском сликом:

1. Позадинска слика се помера са остатком садржаја на страни
2. Позадинска слика се не понавља у оквиру елемента
3. Позадинска слика је непомична (фиксирана) у односу на остатак садржаја
4. Позадинска слика се понавља унутар елемента

2

364. Које од понуђених опција одговара својству: a:visited:

1. стилизовање активног линка
2. стилизовање посећеног линка
3. стилизовање линка које ће се применити када се пређе мишем преко њега
4. стилизовање елемент са називом класе: а

1

365. Дат је следећи код који стилизује три блока – елемента странице са различитим позадинским бојама:

```
<div style="background-color:red;
 width:300px;
 height:100px;
 position:relative;
 top:10px;
 left:80px;
 z-index:2">
</div>

<div style="background-color:yellow;
 width:300px;
 height:100px;
 position:relative;
 top:-60px;
 left:35px;
 z-index:1;">
</div>

<div style="background-color:green;
 width:300px;
 height:100px;
 position:relative;
 top:-220px;
 left:120px;
 z-index:3;">
</div>
```

2

Анализирати и заокружити која од понуђених опција описује горњи код:

1. Слој са зеленом позадином је на врху и преклапа остале слојеве
2. Слој са жутом позадином је на врху и преклапа остале слојеве
3. Слој са црвеном позадином је на врху и преклапа остале слојеве
4. Слојеви се утапају у једну боју – нијансу смеће

366. Заокружити број испред тачног одговора. Која од понуђених опција омогућава постизање ефекта слојевитости елемената у оквиру HTML стране?

1. Својство overlap
2. Својство layer
3. Својство z-index
4. Не постоји решење за постизање слојевитости елемената HTML стране

1

367. Анализирати следећи HTML код и одабрати који од понуђених опција описује резултат приказа HTML кода:

<b><i> Maturski ispit </i></b>

1. Текст ће бити само подебљан
2. Текст ће бити само искошен
3. Текст ће бити исписан подебљано и икошено
4. Неће се применити никакав ефекат на текст

1

### У следећим задацима заокружите бројеве испред тражених одговора

368. Заокружите тачне одговоре. Интернет је:

1. Интернет је светски систем умрежених рачунарских мрежа
2. Софтвер за преглед и приказ www страница се сматра Интернетом
3. Подаци који „путују“ светском мрежом и скуп корисника заједно чине Интернет мрежу
4. Интернет чини њена хардверска компонента као и систем софтверских слојева који контролишу различите аспекте њене комуникационе инфраструктуре

2

369. Међу понуђеним особинама заокружити оне које се односе на динамичке веб странице:

1. Могу приступити базама података
2. Странице се састоје искључиво HTML кода
3. Динамичке странице се пишу у CSS језику
4. Могу слати персонализован садржај појединачним корисницима
5. Странице се извршавају на веб серверу, а резултат овог извршавања представља HTML код
6. Ажирање података на сајту је компликованије и спорије него код статичког сајта

3

370. Заокружити бројеве испред тражених одговора.

Веб сервери са специјализованим програмима су:

1. Microsoft Internet Information Services
2. Microsoft SQL Server
3. Apache Web Server
4. Microsoft NT Server

2

371. Заокружити који су начини за укључивање CSS стила у оквиру HTML стране:

1. Увежена екстерна CSS датотека преко хипервезе у заглављу HTML веб странице - External Style Sheet
2. Дефинисан екстерни CSS стил приказивања у прегледачу HTML страница – CSS plugin
3. Дефинисан интерни CSS стил у оквиру head секције `<style type="text/css">...</style>` - Internal Style Sheet
4. Дефинисан стил у оквиру елемента HTML стране - Inline style
5. Дефинисан стил на крају HTML стране изван head и body секције – Outline style
6. Дефинисан CSS стил у оквиру секције body, унутар тага `<css></css>`

3

372. Заокружити бројеве испред тражених одговора.

Појединачне HTML странице које израђујемо у програму за израду интернет презентација:

1. Можемо приказати одмах и у алату за израду интернет презентација
2. Можемо је приказати само ако се постави на веб сервер
3. Можемо приказати одмах путем веб прегледача са локалног рачунара – без постављања на веб сервер
4. Не може се приказати док се не заврши комплетан веб сајт, постави на веб сервер и прође сигурносне провере веб сервера

1

373. Који су неопходни подаци које треба знати за успешно конфигурисање FTP клијента приликом постављања HTML веб презентације на веб сервер:

1. Име FTP сервера или веб локације
2. Назив интернет опслуживача – провајдера који је обезбедио веб сервер
3. Корисничко име
4. Лозинка
5. IP адреса корисника
6. Назив сервера базе података

1,5

### Допуните следеће реченице и табеле

374. Написати линију кода којом се укључује екстерна CSS датотека - style.css у оквиру заглавља веб странице - index.html (датотеке style.css и indeks.html се налазе у истом директоријуму):

3

### У следећим задацима уредите и повежите појмове према захтеву

375. Повезати појмове према захтеву. Са леве стране су дати интернет сервиси, а са десне стране су дефинисане операције које се могу извршити помоћу тих сервиса.

На линију испред дефинисане операције, написати број њему одговарајућег сервиса.

- |           |       |                                                       |   |
|-----------|-------|-------------------------------------------------------|---|
| 1. E-mail | _____ | успостављање везе са удаљеним рачунаром и рад на њему | 2 |
| 2. FTP    | _____ | приказ HTML веб страница                              |   |
| 3. WWW    | _____ | слanje електронске поште                              |   |
| 4. Telnet | _____ | пренос датотека са удаљеног сервера                   |   |

376. Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML тагови повезују са одговарајућим индикаторима који ће се видети на HTML страници:

- |                           |       |                             |   |
|---------------------------|-------|-----------------------------|---|
| 1. <font color="#ff0003"> | _____ | приказ назива трећег нивоа  | 2 |
| 2. <b>                    | _____ | приказ текста дате величине |   |
| 3. <h3>                   | _____ | приказ текста у боји        |   |
| 4. <font size=„3“>        | _____ | приказ подебљаног текста    |   |

377. Дат је низ HTML наредби. Уписивањем редног броја на предвиђену линију (почев од броја 1 до броја 6) поређајте тачним редоследом HTML наредбе у складу са основном структуром HTML странице:

\_\_\_\_\_ </body>  
 \_\_\_\_\_ <head>  
 \_\_\_\_\_ </html>  
 \_\_\_\_\_ <html>  
 \_\_\_\_\_ </head>  
 \_\_\_\_\_ <body>

3

378. Направите редослед потребних корака за приказ статичке веб странице уносом редних бројева на линије испред описа корака почев од броја 1 до броја 5.

\_\_\_\_\_ Веб сервер шаље пронађену страну клијенту - веб претраживачу.  
 \_\_\_\_\_ Корисник захтева да види веб страну (кликом на линк, укуцањем адресе у адресну линију Веб претраживача, и слично)  
 \_\_\_\_\_ Аутор је креирао страну која се састоји од HTML кода и ставио је на веб сервер.  
 \_\_\_\_\_ Веб претраживач обрађује добијени HTML код и приказује кориснику уредно форматирану страницу са свим елементима (сликама, линковима, табелама, ...)  
 \_\_\_\_\_ Веб сервер проналази HTML страницу коју је корисник захтевао

4

379. Одредите редослед потребних корака за приказ динамичке веб странице уносом редних бројева (почев од 1) на линије испред описа корака.

\_\_\_\_\_ Веб претраживач обрађује добијени HTML код и приказује страницу кориснику са свим елементима.  
 \_\_\_\_\_ Веб сервер извршава програмски код, који је саставни део странице и креира HTML код.  
 \_\_\_\_\_ Корисник креира захтев да виде динамичку веб страницу. Захтев се прослеђује од клијента (веб претраживача) до веб сервера, на којем се налази захтевана страница.  
 \_\_\_\_\_ Аутор је креирао страницу, која се састоји од серверских контрола и инструкција у неком програмском језику, и ставио је на веб сервер.  
 \_\_\_\_\_ Веб сервер обрађује захтев корисника и проналази динамичку страницу коју је корисник захтевао  
 \_\_\_\_\_ Веб сервер шаље преко Интернета генерисани HTML код веб претраживачу.

4

380. Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML тагови повезују са одговарајућим индикаторима који ће се видети на HTML страници:

1. <ol> \_\_\_\_\_ дефинисање елемента листе
2. <li> \_\_\_\_\_ постављање позадинске боје
3. <body bgcolor="#ffff00"> \_\_\_\_\_ постављање позадинске слике
4. <body background="0001.jpg"> \_\_\_\_\_ дефинисање нумерисане листе

2

381. Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML тагови повезују са одговарајућим индикаторима који ће се видети на HTML страници:

- |                           |                             |
|---------------------------|-----------------------------|
| 1. <tr>                   | _____ Дефинисање хиперлинка |
| 2. <td>                   | _____ Нови ред у ћелији     |
| 3.  | _____ Нова ћелија у табели  |
| 4. <a href="index.html" > | _____ Уметање слике         |

2

382. HTML документ може да прими податке од корисника помоћу форми (формулара). Повежите дате тагове и атрибуте са њиховим дефиницијама:

- |           |                                                                                                           |
|-----------|-----------------------------------------------------------------------------------------------------------|
| 1. FORM   | _____ Дефинише одредиште, фајл и који, или е-маил адресу на коју ће подаци са форме бити прослеђени       |
| 2. INPUT  | _____ Одређује начин на који се подаци са форме шаљу на дефинисано одредиште (може бити „post“ или „get“) |
| 3. ACTION | _____ Основни таг формулара са којим се креира формулар за унос података од стране корисника              |
| 4. METHOD | _____ Дефинише поље за унос податка унутар HTML форме.                                                    |

2

## ВЕБ ПРОГРАМИРАЊЕ

### У следећим задацима заокружите број испред траженог одговора

383. Заокружити који протокол TCP/IP стек протокола подржава захтеве веб прегледача: 1
1. FTP
  2. HTTP
  3. TCP
  4. IP
384. Заокружити исказ који описује улогу Domain Name Server-a: 1
1. превођење имена домена у IP адресу
  2. хостовање веб сајта
  3. да буде главни чвор у локалној рачунарској мрежи
  4. приказ динамичких веб страница
385. Једну од платформи за развој веб апликација развио је и Microsoft. Заокружити назив Microsoft-ове платформе за развој веб апликација: 1
1. HTML
  2. JSP
  3. PHP
  4. ASP.NET
386. Заокружите понуђени одговор који одређује шта представља .NET Framework: 1
1. Програмски језик
  2. Библиотека класа
  3. Скрипт језик који се извршава на клијент страни
  4. Скрипт језик који се извршава на серверској страни
387. Заокружити одговор који дефинише улогу Proxy сервера: 2
1. Приступ удаљеном рачунару
  2. Побољшава перформансе конекције, филтрира захтеве и прослеђује их на прави сервер
  3. Пружа хостинг различитим медијским садржајима (Аудио, Видео)
  4. Хостује веб стране
388. Заокружити одговор који дефинише улогу сервера датотека – File server (FTP): 1
1. Омогућавају испоруку веб страна на захтев корисника
  2. Размену електронске поште међу корисницима
  3. Сигуран и поуздан трансфер података са једног рачунара на други
  4. Размену порука у реалном времену

389.	Заокружити одговор који описује шта омогућавају Telnet сервери:	
	1. Размену порука у реалном времену 2. Приступ удаљеном рачунару 3. Размену електронске поште међу корисницима 4. Омогућавају испоруку веб страна на захтев корисника	1
390.	Током извршавања апликације у JavaScript језику, на типове промељивих се односи следећа тврдња (заокружити тачну тврдњу):	
	1. Није могуће мењати типове променљивих у току извршавања апликације 2. Типови променљивих се могу мењати током извршавања програма 3. Сваки тип променљивих се обавезно мења током извршавања апликације у одговарајућу један веб тип премљиве 4. JavaScript не подржава типове промељивих	2
391.	ASP.NET MVC 3.0 долази са новом техником за дефинисање погледа (View Engine). Заокружити назив ове технике:	
	1. ASP.NET View Engine 2. Salome 3. Razor 4. Default	1
392.	Дата је ASP.NET MVC апликација у којој је креирана нова мастер страница (master layout page) која се зове _Layout.WindowsPhone.cshtml. Ако желимо да укључимо нову мастер страницу на новом погледу (View) који сегмент кода треба да искористимо. Заокружити тачан одговор:	
	1. @Html.ActionLink("_Layout.WindowsPhone.cshtml"); 2. Layout = "~/views/Shared/_layout.WindowsPhone.cshtml"; 3. Layout = "Layout.WindowsPhone.cshtml"; 4. @Html.Partial("_Layout.WindowsPhone.cshtml");	2
393.	Зокружити од понуђених опција који симбол се користи за коментаре у ASP.NET MVC Razor синтакси:	
	1. // 2. /* ... */ 3. <!--...--> 4. @* ... *@	1
394.	SOAP протокол (Simple Object Access protocol) који се користи за размену података између рачунара коришћењем веб сервиса у основи користи један скрипт језик. Заокружите од понуђених одговора о којем језику се ради.	
	1. HTML 2. CSS 3. JavaScript 4. XML	1

395.	Од понуђене синтаксе која је исправна формулатија која дефинише верзију XML документа (заокружити један од понуђених одговора):	1
	1. <?xml version="1.0" />	
	2. <xml version="1.0" />	
	3. <?xml version="1.0"?>	
	4. <xml version="1.0"?>	
396.	Заокружити један од понуђених типова сервиса који може бити хостован у конзолној или десктоп апликацији:	2
	1. ASMX	
	2. RESTful	
	3. WCF	
	4. XML	
<b>У следећим задацима заокружите бројеве испред тражених одговора</b>		
397.	Заокружите серверске скрипт језике којима се израђују динамичке веб странице:	1,5
	1. PHP	
	2. JavaScript	
	3. ASP.NET	
	4. JSP	
	5. VBScript	
	6. HTML	
398.	Microsoft .NET Framework садржи базне класе које пружају широк спектар могућности. Заокружити елементе који су укључени у .NET Framework:	3
	1. класе корисничког интерфејса	
	2. класе за приступ подацима и базама	
	3. класе корисника	
	4. веб сервер и примере базе података	
	5. класе за манипулацију XML докумената	
	6. Скрипт језик који се извршава на клијент страни	
399.	У клијентском скрипту језику Java Script, постоји неколико могућности конверзије стринга у број. Заокружити функције које омогућавају те конверзије:	1,5
	1. Није могуће мењати типове променљивих у току извршавања апликације	
	2. Функција EVAL - процењује стринг и ако је могуће претвара га у број	
	3. Функција parseINT – конвертује стринг у целобројни број, ако је могуће	
	4. Функција parseFLOAT – конвертује стринг у реалан број, ако је могуће	
	5. Функција parseDOUBLE – конвертује стринг у реалан број, ако је могуће	
	6. Функција tryParseINT – конвертује стринг у целобројни број, ако је могуће	

<p>400. Повратне вредности акције MVC контролера (controller action method) могу бити различитих типова. Заокружити тачне повратне вредности.</p> <ol style="list-style-type: none"> <li>1. ViewResult</li> <li>2. MVCResult</li> <li>3. ModelResult</li> <li>4. JsonResult</li> <li>5. RedirectResult</li> <li>6. ActionResult</li> </ol>	3
<p>401. Заокружити од понуђених појмова шта дефинише XML Schema (XSD) (заокружити више понуђених одговора):</p> <ol style="list-style-type: none"> <li>1. Типове података XML елемената и атрибута</li> <li>2. Вредности XML елемената и атрибута</li> <li>3. XML елементе који представљају child (деца) елементе</li> <li>4. Уређење child елемената</li> <li>5. Начин сортирања атрибута у оквиру елемената</li> <li>6. Међусобни однос корених (root) елемената документа</li> </ol>	3
<p>402. Заокружити од понуђених појмова шта дефинише језик WSDL (Web Services Description) Language (заокружити више понуђених одговора):</p> <ol style="list-style-type: none"> <li>1. Комуникациони интерфејс за веб сервис</li> <li>2. Начин имплементације метода веб сервиса</li> <li>3. Списак метода веб сервиса</li> <li>4. Комуникациони протокол за веб сервис</li> </ol>	2

### Допуните следеће реченице и табеле

<p>403. Дата је MVC стандардна пута (default route)</p> <p style="text-align: center;"><a href="http://localhost/Customer/Details/5">http://localhost/Customer/Details/5</a></p> <p>која има 3 сегмента. Препознати на основу дате руте вредности ових сегмената и допунити реченицу:</p> <p>Име контролера (Controller Name) је: _____, назив методе (Action Method Name) је: _____ а ID параметра методе је дат са: _____.</p>	1,5
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----

### У следећим задацима уредите и повежите појмове према захтеву

<p>404. У JavaScript језику, многи објекти имају уграђене функције (методе) које симулирају догађаје, ти догађаји настају услед акција корисника. Повежите догађаје са десне стране са одговарајућим акцијама корисника.</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">1. focus</td><td style="width: 30%; text-align: center;">_____</td><td style="width: 40%;">Изађе из фокуса елемента форме</td></tr> <tr> <td>2. submit</td><td style="text-align: center;">_____</td><td>Учита страницу у прегледач</td></tr> <tr> <td>3. load</td><td style="text-align: center;">_____</td><td>Уђе у фокус неког елемента форме</td></tr> <tr> <td>4. blur</td><td style="text-align: center;">_____</td><td>Изврши слање форме</td></tr> </table>	1. focus	_____	Изађе из фокуса елемента форме	2. submit	_____	Учита страницу у прегледач	3. load	_____	Уђе у фокус неког елемента форме	4. blur	_____	Изврши слање форме	2
1. focus	_____	Изађе из фокуса елемента форме											
2. submit	_____	Учита страницу у прегледач											
3. load	_____	Уђе у фокус неког елемента форме											
4. blur	_____	Изврши слање форме											

405.	У JavaScript језику, свака веб страница има доле наведене објекте. Са леве стране су дати називи објеката, а са десне стране својства. Повезати својства објекта са називима објекта:	
1.	window	Садржи својства претходно посетењених URL
2.	location	Садржи својства садржаја текућег документа, као што су назив, боја подлоге, и форме
3.	history	Садржи својства текуће URL
4.	document	Објекат вишег нивоа, садржи својства применљива на цео прозор
406.	Веб обрасци могу да садрже неколико различитих типова компоненти. Категоришице наведене компоненте веб образца и повежите их са датим називима категорија:	2
1.	HTML контроле	TextBox, Label, Button, ListBox, DropDownList, DataGrid
2.	Контроле за податке	FileSystemWatcher, EventLog, MessageQueue
3.	Серверске контроле	SqlConnection, SqlCommand, OleDbConnection
4.	Системске компоненте	Text Area, Table, Image, Submit Button, Reset Button
407.	Садашње верзије IIS сервера изграђене су на модуларној архитектури. Повежите називе модула са функцијама које обављају:	4
1.	HTTP модул	Модул за обављање послова везаних за евиденцију и дијагностику у захтеву
2.	Безбедностни модул	Модул за обављање послова у вези са садржајем у захтеву, као што је обрада захтева за статичке фајлове, враћање подразумевне странице када клиент не наведе ресурс у захтеву и наводи садржај директоријума
3.	Модули за евиденцију и дијагностику	Модул за обављање послова који су специфични за обраду и процесирање захтева, као што је одговорање на информације, враћање HTTP грешака и преусмеравање захтева
4.	Модули садржаја	Модул за обављање послова везаних за безбедност захтева и процесирања, као што су обављање ауторизације УРЛ-а и филтрирање захтева

## ИНФОРМАЦИОНИ СИСТЕМИ И БАЗЕ ПОДАТКА

### У следећим задацима заокружите број испред траженог одговора

408. Заокружити својство података које база података НЕ МОЖЕ да обезбеди: 1
1. Исправност
  2. Истинитост
  3. Потпуност
  4. Заштићеност
- 
409. Једна од функција система за управљање базом података је ауторизација. Означити шта она регулише: 1
1. Права надокнаде за коришћење базе података.
  2. Права приступа појединих корисника појединим објектима (подацима и другим ресурсима) као и права извршења разних операција над тим објектима.
  3. Ауторска права над базом података и програмима који користе ту базу података.
- 
410. Ниво софтвера између корисника и физичких података у бази који штити кориснике базе од детаља на физичком нивоу назива се: 1
1. Програмски језик
  2. Систем за управљање базама података
  3. База података
  4. Case алат
- 
411. Посматрајмо релациони модел БИБЛИОТЕКЕ. Заокружити ком делу релационог модела припада услов да два издавача не могу имати исти назив: 2
1. Структурном делу
  2. Манипулативном делу
  3. Интегративном делу
  4. Софтверском делу
- 
412. Међу понуђеним, одредити ентитет са атрибутима који НИСУ одговарајући: 2
1. Ентитет: СТУДЕНТ – Атрибути: име, презиме, смер, број бодова, просек
  2. Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина
  3. Ентитет: КЊИГА – Атрибути: наслов, аутор, година издања, издавач, адреса издавача, телефон издавача
  4. Ентитет: АВИОН – Атрибути: произвођач, марка, година производње, број седишта
- 
413. Сврха јединственог идентификатора (кључа) је да: 1
1. Јединствено одређује колону те табеле.
  2. Јединствено одређује табелу и врсту табеле.
  3. Јединствено одређује инстанцу ентитета.
  4. Јединствено одређује ентитет у бази података.

414.	Заокружити тачан исказ:	
	1. Кардиналност неке везе представља однос броја објеката који се повезују. 2. Кардиналност неке везе представља апстракцију у којој се скуп сличних типова објеката представља општим генеричким типом (надтиповим). 3. Кардиналност неке везе одређује опционалност учешћа у вези.	2
415.	Заокружите тачан исказ:	
	1. Примарни кључ је атрибут који мора бити целобројног типа. 2. Примарни кључ је атрибут који указује на зависност од неке друге табеле. 3. Примарни кључ је атрибут који јединствено идентификује врсте у табели. 4. Ако табела садржи вишевредносни атрибут, њему се додељује функција примарног кључа.	1
416.	Заокружити број испред исказа који описује шта резервисана реч DISTINCT проузрокује у SELECT наредби:	
	1. Не приказује NULL вредности. 2. Елиминише све јединствене вредности из табеле. 3. Елиминише све врсте које се више пута јављају у приказу. 4. Елиминише само јединствене врсте у приказу.	1
417.	Одредити оператор који би требало користити у SELECT наредби да би били приказани само они ученици чије презиме почиње словом A:	
	1. IN 2. LIKE 3. BETWEEN 4. IS LIKE 5. BEGINS WITH	2
418.	Одредити оператор који би требало употребити у <b>SELECT</b> наредби да би биле приказане све врсте табеле <b>UCENICI</b> које садрже <b>null</b> вредности у колони <b>stipendija_id</b> :	
	1. “=NULL” 2. ISNULL 3. ==NULL 4. IS NULL 5. LIKE NULL	2
419.	Одредити коју клаузулу је потребно додати упиту	
	<pre>SELECT prezime, ime, skola_id, mentor_id FROM takmicari</pre>	
	да би такмичари били уређени најпре по школи, а онда ментору, па по презимену ученика:	1
	1. ORDER BY skola_id, prezime 2. ORDER BY skola_id, mentor_id, prezime, ime 3. ORDER BY skola_id, mentor_id, prezime 4. ORDER BY prezime, mentor_id, skola_id	

420. Извршава се упит:

```
SELECT prezime, ime, email
FROM ucenici
ORDER BY prezime DESC
```

Заокружити исказ који је истинит:

1. Све врсте ће бити сортиране по презимену у абецедном поретку.
2. Све врсте ће бити сортиране по презимену у обрнутом абецедном поретку.
3. Све врсте ће бити сортиране по презимену, затим по имену, а на крају по имејл адреси.

1

421. Одредити резервисану реч коју је потребно укључити у ORDER BY клаузулу да би приказали податке из табеле са подацима о такмичарима уређене у опадајућем поретку по броју бодова:

1. DESC
2. ASC
3. SORT
4. CHANGE
5. SORT DESC

1

422. Заокружити број испред понуђеног одговора који представља могући резултат рада следеће SQL наредбе:

```
SELECT ucenik_id
FROM ucenici
WHERE ucenik_id BETWEEN 200 AND 250
OR ucenik_id IN (220, 280, 310)
AND (ucenik_id BETWEEN 250 AND 300)
```

2

1. 240, 250, 260, 280
2. 305, 306, 307, 308, 309
3. 220, 225, 230, 250, 280
4. 120, 180, 310

423. За преглед података из табеле која се налази у оквиру базе података користи се наредба:

1. ADD
2. DROP
3. INSERT
4. SELECT

1

424. За измену структуре табеле која се налази у оквиру базе података (додавање или брисање, тј. уклањање колоне) користи се наредба:

1. SELECT
2. CREATE
3. INSERT
4. ALTER

1

425. Заокружити број испред траженог одговора. За додавање нове колоне у табелу која се налази у оквиру базе података користи се наредба ALTER у комбинацији са:

1. ADD\_COLUMN
2. NEW\_COLUMN
3. INSERT
4. ADD

1

426. Дата је табела **KUPCI**:

<b><i>Id</i></b>	<b><i>Prezime</i></b>	<b><i>Adresa</i></b>	<b><i>Mesto</i></b>	<b><i>PostBroj</i></b>	<b><i>Status</i></b>
1	Antić	Mokroluška 35	Bečeј	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masarikova 1	Piroт	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečeј	21220	Aktivan

Извршава се упит:

`DELETE FROM KUPCI WHERE Mesto = 'Niš'`

Заокружити број испред траженог одговора. Колико ће записа остати у табели?

1. 0 записа
2. 2 записа
3. 4 записа
4. 3 записа

1

427. Дата је табела **KUPCI**:

<b><i>Id</i></b>	<b><i>Prezime</i></b>	<b><i>Adresa</i></b>	<b><i>Mesto</i></b>	<b><i>PostBroj</i></b>	<b><i>Status</i></b>
1	Antić	Mokroluška 35	Bečeј	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masarikova 1	Piroт	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečeј	21220	Aktivan

Извршава се упит:

`ALTER TABLE KUPCI ADD (DatUgovora (Date))`

2

Оредити шта ће се десити након извршења упита. Заокружити број испред траженог одговора.

1. Промениће се тип података у колони DatUgovora у табели KUPCI
2. Биће додат нови запис – врста у табелу KUPCI
3. Биће додата нова колона под именом DatUgovora у табелу KUPCI
4. Биће преименована колона у табели KUPCI
5. Биће уклоњена – обрисана колона у табели KUPCI

428. Дата је табела **KUPCI**:

<b><i>Id</i></b>	<b><i>Prezime</i></b>	<b><i>Adresa</i></b>	<b><i>Mesto</i></b>	<b><i>PostBroj</i></b>	<b><i>Status</i></b>
1	Antić	Mokroluška 35	Bečeј	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masarikova 1	Piroт	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečeј	21220	Aktivan

Извршава се упит:

`SELECT Mesto, Adresa FROM KUPCI WHERE Prezime LIKE '%tić'`

1

Оредити колико редова ће имати резултантна табела не рачунајући заглавље:

1. 0 редова
2. 1 ред
3. 2 реда
4. 3 реда
5. 6 редова

429. Дата је табела **KUPCI**:

<b><i>Id</i></b>	<b><i>Prezime</i></b>	<b><i>Adresa</i></b>	<b><i>Mesto</i></b>	<b><i>PostBroj</i></b>	<b><i>Status</i></b>
1	Antić	Mokroluška 35	Bečeј	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masaričkova 1	Pirot	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečeј	21220	Aktivan

Извршава се упит:

`SELECT Mesto, Adresa FROM KUPCI WHERE Prezime LIKE '%mić'`

Одредити колико редова ће имати резултантна табела не рачунајући заглавље:

1. 0 редова
2. 1 ред
3. 2 реда
4. 3 реда
5. 6 редова

1

430. Дата је табела **KUPCI**:

<b><i>Id</i></b>	<b><i>Prezime</i></b>	<b><i>Adresa</i></b>	<b><i>Mesto</i></b>	<b><i>PostBroj</i></b>	<b><i>Status</i></b>
1	Antić	Mokroluška 35	Bečeј	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masaričkova 1	Pirot	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečeј	21220	Aktivan

Извршава се упит:

`SELECT DISTINCT Mesto FROM KUPCI`

Одредити колико редова ће имати резултантна табела не рачунајући заглавље:

1. 5 редова
2. 4 реда
3. 2 реда
4. 3 реда
5. 6 редова

2

431. Дата је табела **KUPCI**:

<b><i>Id</i></b>	<b><i>Prezime</i></b>	<b><i>Adresa</i></b>	<b><i>Mesto</i></b>	<b><i>PostBroj</i></b>	<b><i>Status</i></b>
1	Antić	Mokroluška 35	Bečeј	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masaričkova 1	Pirot	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečeј	21220	Aktivan

Извршава се упит:

`SELECT Prezime, Mesto, Adresa FROM KUPCI WHERE LENGTH(Adresa) > 14`

Одредити колико редова ће имати резултантна табела не рачунајући заглавље:

1. 5 редова
2. 4 реда
3. 2 реда
4. 3 реда
5. 0 редова

1

432. Дата је табела **KUPCI**:

<b>Id</b>	<b>Prezime</b>	<b>Adresa</b>	<b>Mesto</b>	<b>PostBroj</b>	<b>Status</b>
1	Antić	Mokroluška 35	Bečej	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masaričkova 1	Pirot	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečej	21220	Aktivan

Извршава се упит:

`SELECT Prezime, Mesto, Adresa FROM KUPCI WHERE LENGTH(Adresa) < 10`

Одредити колико редова ће имати резултантна табела не рачунајући заглавље:

1. 5 редова
2. 4 реда
3. 2 реда
4. 1 ред
5. 0 редова

1

433. Дата је табела **ARTIKLI**:

<b>Id</b>	<b>Naziv</b>	<b>DatProiz</b>	<b>JM</b>	<b>Kol</b>	<b>Cena</b>	<b>Akcija</b>
21	Hleb Sava	15.04.2015	kom	100	35	NULL
32	Hleb Dunav	15.04.2015	kom	50	45	0.05
23	Baget beli	12.04.2015	kom	20	32	0.12
47	Baget crni	12.04.2015	kom	12	35	0.08
25	Tost tamni	10.04.2015	kom	6	75	0.15
36	Tost beli	10.04.2015	kom	8	70	NULL
28	Kifla integralna	15.04.2015	kom	15	18	0.01
10	Brašno Graham	10.03.2015	kg	24	54	0.03

2

Одредити упит који приказује називе за све артикли и за сваки одредити укупну цену за дату количину:

1. `SELECT Naziv, Kol * Cena as Ukupno FROM ARTIKLI`
2. `SELECT Naziv, COUNT(Kol * Cena) FROM ARTIKLI`
3. `SELECT Naziv, COUNT(Kol , Cena) as Ukupno FROM ARTIKLI`
4. `SELECT Naziv, Kol + Cena FROM ARTIKLI`

## 434. Дата је табела ARTIKLI:

<b>Id</b>	<b>Naziv</b>	<b>DatProiz</b>	<b>JM</b>	<b>Kol</b>	<b>Cena</b>	<b>Akcija</b>
21	Hleb Sava	15.04.15	kom	100	35	NULL
32	Hleb Dunav	15.04.15	kom	50	45	0.05
23	Baget beli	12.04.15	kom	20	32	0.12
47	Baget crni	12.04.15	kom	12	35	0.08
25	Tost tamni	10.04.15	kom	6	75	0.15
36	Tost beli	10.04.15	kom	8	70	NULL
28	Kifla integralna	15.04.15	kom	15	18	0.01
10	Brašno Graham	10.03.15	kg	24	54	0.03

3

Одредити упит који приказује називе за све артикли на акцији и нове цене, умањене за акцијски попуст:

1. `SELECT Naziv, Akcija * Cena FROM ARTIKLI where Akcija is not NULL`
2. `SELECT Naziv, (1-Akcija) * Cena FROM ARTIKLI where Akcija is NULL`
3. `SELECT Naziv, Akcija * Cena FROM ARTIKLI where Akcija <> 0`
4. `SELECT Naziv, (1-Akcija) * Cena FROM ARTIKLI where Akcija is not NULL`

435. Дата је табела **ARTIKLI**:

<b>Id</b>	<b>Naziv</b>	<b>DatProiz</b>	<b>JM</b>	<b>Kol</b>	<b>Cena</b>	<b>Akcija</b>
21	Hleb Sava	15.04.15	kom	100	35	NULL
32	Hleb Dunav	15.04.15	kom	50	45	0.05
23	Baget beli	12.04.15	kom	20	32	0.12
47	Baget crni	12.04.15	kom	12	35	0.08
25	Tost tamni	10.04.15	kom	6	75	0.15
36	Tost beli	10.04.15	kom	8	70	NULL
28	Kifla integralna	15.04.15	kom	15	18	0.01
10	Brašno Graham	10.03.15	kg	24	54	0.03

2

Извршава се упит:

`SELECT Naziv, DatProiz, Kol FROM Artikli WHERE Id < 30 AND Cena <= 45`

Одредити колико редова има табела која је резултат рада упита не рачунајући заглавље:

1. 0 редова
2. 1 ред
3. 2 реда
4. 3 реда

436. Дата је табела **ARTIKLI**:

<b>Id</b>	<b>Naziv</b>	<b>DatProiz</b>	<b>JM</b>	<b>Kol</b>	<b>Cena</b>	<b>Akcija</b>
23	Baget beli	12.04.15	kom	20	32	0.12
47	Baget crni	12.04.15	kom	12	35	0.08
25	Tost tamni	10.04.15	kom	6	75	0.15
36	Tost beli	10.04.15	kom	8	70	NULL
28	Kifla integralna	15.04.15	kom	15	18	0.01
10	Brašno Graham	10.03.15	kg	24	54	0.03

1

Извршава се упит:

`SELECT MAX(Cena), MIN(Cena) FROM Artikli`

Одредити резултат рада упита:

1.	MAX(Cena)	MIN(Cena)
	75	32
2.	MAX(Cena)	MIN(Cena)
	75	18
3.	MIN(Cena)	MAX(Cena)
	6	100
4.	MIN(Cena)	MAX(Cena)
	18	54

437. Одабрати клаузулу коју је потребно користити уколико задатак захтева да SELECT упит врати подскуп података из неке табеле/табела:

1. ANYWHERE
2. WHICH
3. WHERE
4. EVERY

1

438. После извршења наредбе:

`CREATE INDEX ind_ime ON RADNIK (IME ASC)`

1. у табели RADNIK се поставља INDEX над колоном IME
2. у табели INDEX се поставља IME над колоном RADNIK
3. у табели RADNIK се поставља INDEX над колоном ind\_ime
4. у табели RADNIK се поставља INDEX над колоном ASC

1

439. Дата је табела **ARTIKLI**:

<b>Id</b>	<b>Naziv</b>	<b>DatProiz</b>	<b>JM</b>	<b>Kol</b>	<b>Cena</b>	<b>Akcija</b>
21	Hleb Sava	15.04.15	kom	100	35	NULL
32	Hleb Dunav	15.04.15	kom	50	45	0.05
23	Baget beli	12.04.15	kom	20	32	0.12
47	Baget crni	12.04.15	kom	12	35	0.08
25	Tost tamni	10.04.15	kom	6	75	0.15
36	Tost beli	10.04.15	kom	8	70	NULL
28	Kifla integralna	15.04.15	kom	15	18	0.01
10	Brašno Graham	10.03.15	kg	24	54	0.03

2

Извршава се упит:

```
SELECT Naziv, DatProiz, Kol FROM Artikli WHERE Id > 30 AND Cena >= 45
```

Резултат извршења упита је табела са заглављем и

1. 0 редова и 2 колоне
2. 1 редом и 3 колоне
3. 2 реда и 3 колоне
4. 3 реда и 2 колоне

440. Дата је табела **KUPCI**:

<b>Id</b>	<b>Prezime</b>	<b>Adresa</b>	<b>Mesto</b>	<b>PostBroj</b>	<b>Status</b>
1	Antić	Mokroluška 35	Bečej	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masarikova 1	Pirot	18300	Pasivan
4	Perić	Takovska 6	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečej	21220	Aktivan

Извршава се упит:

```
SELECT Prezime, Mesto, Adresa FROM KUPCI WHERE LENGTH(Adresa) > 10
```

1

Заокружити број испред траженог одговора. Резултат извршења упита је табела са заглављем и:

1. 5 редова и 3 колоне
2. 4 реда и 4 колоне
3. 2 реда и 3 колоне
4. 1 редом и 2 колоне
5. 0 редова и 3 колоне

441. Дата је табела **KUPCI**:

<b>Id</b>	<b>Prezime</b>	<b>Adresa</b>	<b>Mesto</b>	<b>PostBroj</b>	<b>Status</b>
1	Antić	Mokroluška 35	Bečej	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masarikova 1	Pirot	18300	Pasivan
4	Perić	Takovska 6	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečej	21220	Aktivan

1

Извршава се упит:

```
SELECT Prezime, Adresa, Mesto FROM KUPCI ORDER BY Prezime
```

Записи у резултантној табели ће имати:

1. поредак исти као у приказаној табели
2. поредак супротан од поретка у приказаној табели
3. поредак изменјен у односу на приказану табелу, али не супротан

442. Дата је табела **KUPCI**:

<b><i>Id</i></b>	<b><i>Prezime</i></b>	<b><i>Adresa</i></b>	<b><i>Mesto</i></b>	<b><i>PostBroj</i></b>	<b><i>Status</i></b>
1	Antić	Mokroluška 35	Bečeј	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masarikova 1	Pirot	18300	Pasivan
4	Perić	Takovska 6	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečeј	21220	Aktivan

Извршава се упит:

`SELECT COUNT(*) FROM KUPCI WHERE Status= 'Aktivan'`

Одредити резултат упита:

1. Резултат упита је 3
2. Резултат упита је 4
3. Резултат упита је 2
4. Резултат упита је 1
5. Резултат упита је 0

1

443. Дата је табела **KUPCI**:

<b><i>Id</i></b>	<b><i>Prezime</i></b>	<b><i>Adresa</i></b>	<b><i>Mesto</i></b>	<b><i>PostBroj</i></b>	<b><i>Status</i></b>
1	Antić	Mokroluška 35	Bečeј	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masarikova 1	Pirot	18300	Pasivan
4	Perić	Takovska 6	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečeј	21220	Aktivan

Извршава се упит:

`SELECT Prezime, Adresa, Mesto FROM KUPCI ORDER BY Mesto`

Записи у резултантној табели ће имати:

1. поредак исти као у приказаној табели
2. поредак супротан од приказане табеле
3. поредак изменењен у односу на приказану табелу, али не супротан

1

444. После извршења дате наредбе:

```
CREATE TABLE RADNIK
(
 IDBR INTEGER,
 IME VARCHAR(20) NOT NULL,
 POSAO VARCHAR(15) NOT NULL,
 KVALIF VARCHAR(3),
 RUKOVODILAC INTEGER,
 DATZAP DATE,
 PREMIJA FLOAT,
 PLATA FLOAT,
 BROD INTEGER
)
```

1

добија се:

1. табела RADNIK са девет колона од којих се четири обавезно попуњавају
2. табела RADNIK са шест колона од којих се три обавезно попуњавају
3. табела RADNIK са девет колона од којих се две обавезно попуњавају
4. табела RADNIK са осам колона од којих се једна обавезно попуњава

445. После извршења наредбе:

```
CREATE TABLE RADNIK
(IDBR INTEGER,
 IME VARCHAR(20) NOT NULL,
 POSAO VARCHAR(15) NOT NULL,
 KVALIF VARCHAR(3),
 RUKOVODILAC INTEGER,
 DATZAP DATE,
 PREMIJA FLOAT,
 PLATA FLOAT,
 BROD INTEGER
)
```

добија се:

1. табела RADNIK са примарним кључем и страним кључем
2. табела RADNIK са примарним кључем и без страног кључа
3. табела RADNIK без примарног кључка и са страним кључем
4. табела RADNIK без примарног кључка и без страног кључа

2

446. Дата је табела **ARTIKLI**:

ID	NAZIV	DATPROIZ	JM	KOL	CENA	AKCIJA
21	Hleb Sava	15.04.15	kom	100	35	NULL
32	Hleb Dunav	15.04.15	kom	50	45	0.05
23	Baget beli	12.04.15	kom	20	32	0.12
47	Baget crni	12.04.15	kom	12	35	0.08
25	Tost tamni	10.04.15	kom	6	75	0.15
36	Tost beli	10.04.15	kom	8	70	NULL
28	Kifla integralna	15.04.15	kom	15	18	0.01
10	Brašno Graham	10.03.15	kg	24	54	0.03

2

Одредити упит који коригује – повећава цене за све артикли линеарно за 8 процената у односу на тренутну цену. Заокружити број испред траженог одговора:

1. `UPDATE ARTIKLI SET Cena = Cena*0.8`
2. `UPDATE ARTIKLI SET Cena AS Cena*0.8`
3. `UPDATE ARTIKLI SET Cena = Cena*(1+0.08)`
4. `UPDATE ARTIKLI SET Cena AS Cena*(1 + 0.08)`

447. Дата је табела **RADNIK**:

IDBR	IME	PREZIME	PLATA	PREMIJA	DATZAP
5874	Todor	Marjanović	1000	1100	19.4.1971
5898	Andrija	Aleksić	1100	NULL	20.1.1980
5900	Slobodan	Golubović	900	1300	3.10.1978
5932	Mitar	Gavrilović	2600	NULL	25.3.1985
5953	Persida	Kosanović	1100	0	12.1.1979
6234	Marko	Pavlović	1300	3000	17.12.1990
6789	Janko	Nikolić	3900	10	23.12.1999

1

Заокружити број испред упита којим ће се приказати само презиме, име и датум запослења за раднике запослене пре 1. јануара 1980.

1. `SELECT Prezime, Ime, DatZap FROM Radnik`
2. `SELECT Ime, Prezime, DatZap FROM Radnik WHERE DatZap < '1.1.1980'`
3. `SELECT Prezime, Ime, DatZap FROM Radnik WHERE DatZap > '1980/1/1'`
4. `SELECT Prezime, Ime, DatZap FROM Radnik WHERE DatZap < '1980-1-1'`

448. Дата је табела **RADNIK**:

IDBR	IME	PREZIME	PLATA	PREMIJA	DATZAP
5900	Slobodan	Golubović	3900	1300	3.10.1978
5932	Mitar	Gavrilović	2600	NULL	25.3.1985
5953	Persida	Kosanović	1100	0	12.1.1979
6234	Marko	Pavlović	1300	3000	17.12.1990
6789	Janko	Nikolić	3900	10	31.12.1995

Извршава се упит:

```
SELECT ime, prezime, plata FROM radnik
WHERE plata = (SELECT MAX(plata) FROM radnik)
```

2

Заокружити број испред одговора који представља резултат извршења упита:

1. 

	IME	PREZIME	PLATA
	Slobodan	Golubović	3900
	Mitar	Gavrilović	2600
2. 

	IME	PREZIME	PLATA
	Slobodan	Golubović	3900
	Janko	Nikolić	3900
3. 

	IME	PREZIME	PLATA
	Slobodan	Golubović	3900
4. 

	IME	PREZIME	PLATA
	Janko	Nikolić	3900

449. Дата је табела **RADNIK**:

IDBR	IME	PREZIME	PLATA	PREMIJA	DATZAP
5900	Slobodan	Golubović	3900	1300	3.10.1978
5932	Mitar	Gavrilović	2600	NULL	25.3.1985
5953	Persida	Kosanović	1100	0	12.1.1979
6234	Marko	Pavlović	1300	3000	17.12.1990
6789	Janko	Nikolić	3900	10	31.12.1995

Извршава се упит:

```
SELECT ime, prezime, plata+premija AS zarada FROM radnik
WHERE premija is not NULL and premija > 10
```

2

Заокружити број испред одговора који представља резултат извршења упита:

1. 

	IME	PREZIME	ZARADA
	Slobodan	Golubović	5200
	Marko	Pavlović	4300
2. 

	IME	PREZIME	ZARADA
	Slobodan	Golubović	3900
	Marko	Pavlović	3000
3. 

	IME	PREZIME	ZARADA
	Mitar	Gavrilović	2600
4. 

	IME	PREZIME	ZARADA
	Persida	Kosanović	1100
	Janko	Nikolić	3900

450. Извршењем дате наредбе:

```
ALTER TABLE PROJEKAT ADD RokKraj date
```

постиже се:

1. у бази података у табели PROJEKAT додаје се ограничење RokKraj
2. у бази података у табели PROJEKAT додаје се колона RokKraj
3. у бази података у табели RokKraj додаје се колона PROJEKAT
4. у базу података додаје се табела PROJEKAT

2

451. Дата је табела **RADNIK**:

IDBR	IME	PREZIME	PLATA	PREMIJA	DATZAP
5900	Slobodan	Golubović	3900	1300	3.10.1978
5932	Mitar	Gavrilović	2600	NULL	25.3.1985
5953	Persida	Kosanović	1100	0	12.1.1979
6234	Marko	Pavlović	1300	3000	17.12.1990
6789	Janko	Nikolić	3900	10	31.12.1995

Извршава се упит:

```
SELECT ime, prezime, plata+premija AS zarada FROM radnik
WHERE premija is not NULL
```

2

Одредити колико редова има табела која је резултат извршења упита не рачунајући заглавље. Заокружити број испред тачног одговора:

1. 0 реда
2. 2 реда
3. 4 реда
4. 5 редова

452. Дата је табела **RADNIK** и табела **ODELJENJE** које су међусобно повезане пољем **BROD** (број одељења).

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	3900	10
5932	Mitar	Gavrilović	2600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	NULL
6789	Janko	Nikolić	3900	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

Извршава се упит:

```
SELECT Naziv
FROM Odeljenje
WHERE Brod NOT IN (SELECT Brod FROM Radnik)
```

2

Одредити шта се приказује након извршења упита. Заокружити број испред траженог одговора:

1. Називи свих одељења
2. Називи одељења у којима су распоређени радници
3. Називи одељења у којима нису распоређени радници
4. Назив одељења у ком има највише радника

453. Дата је табела **KUPCI**:

<b>Id</b>	<b>Prezime</b>	<b>Adresa</b>	<b>Mesto</b>	<b>PostBroj</b>	<b>Status</b>
1	Antić	Mokroluška 35	Bečej	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masaričkova 1	Pirot	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečej	21220	Aktivan

Извршава се упит:

`DELETE FROM KUPCI WHERE Mesto LIKE '%_o_%'`

2

Одредити вредности **Id** оних записа који ће бити уклоњени из табеле након извршења датог упита. Заокружити број испред траженог одговора:

1. 1, 2
2. 2, 3
3. 3, 5
4. 1, 6

454. Дата је табела **KUPCI** у којој поља **Adresa**, **PostBroj**, **Status** могу бити неодређена.

<b>Id</b>	<b>Prezime</b>	<b>Adresa</b>	<b>Mesto</b>	<b>PostBroj</b>	<b>Status</b>
1	Antić	Mokroluška 35	Bečej	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masaričkova 1	Pirot	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečej	21220	Aktivan

Извршава се упит:

1

`INSERT INTO KUPCI (Id, Prezime, Mesto)  
VALUES (9, 'Andrejić', 'Ada')`

Одредити резултат рада овог упита. Заокружити број испред траженог одговора:

1. Добићемо поруку о грешци
2. У табелу ће бити додат један запис
3. Табела се неће променити
4. Додате вредности нису правилне

455. Дата је табела **KUPCI**:

<b>Id</b>	<b>Prezime</b>	<b>Adresa</b>	<b>Mesto</b>	<b>PostBroj</b>	<b>Status</b>
1	Antić	Mokroluška 35	Bečej	21220	Aktivan
2	Kostić	Jug Bogdanova 12	Niš	18000	Pasivan
3	Marić	Masaričkova 1	Pirot	18300	Pasivan
4	Perić	Takovska 67	Niš	18000	Pasivan
5	Rakić	Konstantinova 22	Sombor	25000	Aktivan
6	Tomić	Kneza Miloša 13	Bečej	21220	Aktivan

Извршава се упит:

1

`CREATE TABLE NoviKupci(Id integer primary key, Prezime varchar(50), Adresa varchar(50), Mesto varchar(20), PostBroj varchar(5), Status varchar(8))`

Одредити резултат извршења упита. Заокружити број испред траженог одговора:

1. Креира нову табелу NoviKupci
2. У нову табелу NoviKupci преписује све записи из табеле KUPCI
3. Уклања табелу KUPCI и замењује је табелом NoviKupci
4. Не креира нову табелу NoviKupci јер већ постоји табела KUPCI

456. Дате су табеле: **Kupci**, **NoviKupci** следеће (једнаке) структуре:

```
(Id integer primary key, Prezime varchar(50), Adresa varchar(50),
Mesto varchar(20), PostBroj varchar(5), Status varchar(8))
```

Извршава се упит:

```
INSERT INTO NoviKupci
SELECT * FROM Kupci WHERE Status = 'Aktivan'
```

2

Заоружити број испред одговора који представља резултат извршења упита:

1. У табелу **NoviKupci** уписује све записи из табеле **Kupci**
2. У табелу **NoviKupci** уписује само записи из табеле **Kupci** са задатим статусом
3. У табелу **Kupci** уписује све записи из табеле **NoviKupci**
4. У табелу **Kupci** уписује само записи из табеле **NoviKupci** са задатим статусом

457. Дате су табеле: **KUPCI**, **NoviKupci** следеће (једнаке) структуре:

```
(Id integer primary key, Prezime varchar(50), Adresa varchar(50), Mesto
varchar(20), PostBroj varchar(5), Status varchar(8))
```

Извршава се упит:

```
INSERT INTO NoviKupci
SELECT * FROM Kupci WHERE Status <> 'Aktivan'
```

2

Одредити шта је резултат извршења датог упита. Заокружити број испред траженог одговора:

1. У табелу **NoviKupci** уписује само записи из табеле **Kupci** са статусом који је једнак **Aktivan**
2. У табелу **NoviKupci** уписује само записи из табеле **Kupci** са статусом који није једнак **Aktivan**
3. У табелу **Kupci** уписује све записи из табеле **NoviKupci**
4. У табелу **Kupci** уписује само записи из табеле **NoviKupci** са статусом који није једнак **Aktivan**

458. Дата је табела **ARTIKLI**:

Id	Naziv	DatProiz	JM	Kol	Cena	Aкција
21	Hleb Sava	15.04.2015	kom	100	35	NULL
32	Hleb Dunav	15.04.2015	kom	50	45	0.05
23	Baget beli	12.04.2015	kom	20	32	0.12
47	Baget crni	12.04.2015	kom	12	35	0.08
25	Tost tamni	10.04.2015	kom	6	75	0.15
36	Tost beli	10.04.2015	kom	8	70	NULL
28	Kifla integralna	15.04.2015	kom	15	18	0.01
10	Brašno Graham	10.03.2015	kg	24	54	0.05

2

Потребно је приказати називе, цене и количине за све артикли чији назив **не почиње** са Hleb, Baget или Tost. Заокружити број испред упита који одговара постављеном захтеву.

1. `SELECT Naziv, Kol, Cena FROM Artikli`
2. `SELECT Naziv, Kol, Cena FROM Artikli WHERE Naziv NOT IN ('Hleb%', 'Baget%', 'Tost%')`
3. `SELECT Naziv, Kol, Cena FROM Artikli WHERE Naziv NOT LIKE 'Hleb%' AND Naziv NOT LIKE 'Tost%' AND Naziv NOT LIKE 'Baget%'`
4. `SELECT Naziv, Kol, Cena FROM Artikli WHERE Naziv NOT LIKE ('Hleb%', 'Baget%', 'Tost%')`

459. Дата је табела **ARTIKLI** и упит:

<b>Id</b>	<b>Naziv</b>	<b>DatProiz</b>	<b>JM</b>	<b>Kol</b>	<b>Cena</b>	<b>Akcija</b>
21	Hleb Sava	15.04.2015	kom	100	35	NULL
32	Hleb Dunav	15.04.2015	kom	50	45	0.05
23	Baget beli	12.04.2015	kom	20	32	0.12
47	Baget crni	12.04.2015	kom	12	35	0.08
36	Tost beli	10.04.2015	kom	8	70	NULL
28	Kifla integralna	15.04.2015	kom	15	18	0.01
10	Brašno Graham	10.03.2015	kg	24	54	0.03

**SELECT Naziv, Kol, Cena, DATEDIFF("d", DatProiz, '2015-04-18') FROM Artikli**

2

Одредити шта ће бити приказано након извршења упита. Заокружити број испред траженог одговора:

1. Називи, цене и количине за све артикли
2. Називи, цене и количине за све артикле који су произведени пре 18.04.2015.
3. Називи, цене, количине и број дана од датума производње до 18.04.2015. за све артикле
4. Називи, цене, количине и број дана од датума производње до дана истека рока за све артикле

460. Дата је табела **RADNIK** и упит:

<b>IDBR</b>	<b>IME</b>	<b>PREZIME</b>	<b>PLATA</b>	<b>PREMIJA</b>	<b>DATZAP</b>
6234	Marko	Pavlović	1300	3000	17.12.1990
6789	Janko	Nikolić	3900	10	23.12.1999

**SELECT Ime, Prezime, DATEDIFF("yyyy", DatZap, GETDATE()) AS God FROM Radnik**

2

Одредити шта је резултат упита. Заокружити број испред траженог одговора:

1. Табела са подацима о именима и презименима радника
2. Табела са подацима о именима, презименима и броју година које су протекле од датума запослења радника до краја века
3. Табела са подацима о именима, презименима и датумима запослења радника
4. Табела са подацима о именима, презименима и броју година које су протекле од датума запослења радника до тренутног датума

461. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

<b>IDBR</b>	<b>IME</b>	<b>PREZIME</b>	<b>PLATA</b>	<b>BROD</b>
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

<b>BROD</b>	<b>Naziv</b>	<b>Mesto</b>
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

**SELECT Imeod, AVG(Plata) AS ProsekPlata FROM Radnik, Odeljenje**

2

**WHERE Odeljenje.Brod=Radnik.Brod GROUP BY Imeod HAVING AVG(Plata)>1000**

Одредити резултат извршавања датог упита. Заокружити број испред траженог одговора:

1. Приказују називи свих одељења и висина просечне плате у њима
2. Групишу по одељењима радници са платом већом од просечне плате
3. Приказују називи одељења и висина просечне плате у њима само за одељења у којима је просечна плата већа од 1000
4. Приказују бројеви свих одељења и висина просечне плате у њима

462. Дата је табела **RADNIK** (*IDBR, Prezime, Ime, Plata, Datzap, BROD*) и упит:

```
SELECT Prezime, Ime, Posao, Plata
FROM Radnik WHERE Ime LIKE @name
```

Одредити групу којој припада дати упит. Заокружити број испред траженог одговора:

1. Параметарски упити
2. Групни упити
3. Акциони упити
4. Агрегатни упити

1

463. Заокружити број испред траженог одговора. Упит:

```
CREATE TABLE NoviKupci(Id integer primary key, Prezime varchar(50),
Adresa varchar(50), Mesto varchar(20), PostBroj varchar(5), Status
varchar(8))
```

спада у:

1. Параметарске упите
2. Акционе упите
3. Селекционе упите
4. Агрегатне упите

1

464. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

Извршава се упит:

```
UPDATE Radnik SET Radnik.brod = 40
WHERE Radnik.brod IS NULL
```

2

Одредите шта се остварује извршењем овог упита. Заокружити број испред траженог одговора:

1. Распоређивање свих радника у одељење чији је број 40
2. Брисање нераспоређених радника
3. Уклањање одељења чији је број 40
4. Распоређивање нераспоређених радника у одељење чији је број 40

465. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

1

Одредити ком типу упита припада упит којим се радници из одељења **Plasman** прераспоређују у одељење **Nabavka**. Заокружити број испред траженог одговора:

1. Параметарски упит
2. Агрегатни упит
3. Угњеждени упит
4. Акциони упит

466. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

Одредити упит којим се радници из одељења **Plasman** прераспоређују у одељење **Nabavka**. Заокружити број испред траженог одговора:

1. **UPDATE Radnik SET brod = 10 WHERE brod=40**
2. **UPDATE Radnik SET brod = 40 WHERE brod=10**
3. **ALTER TABLE Radnik SET brod = 40 WHERE brod=10**
4. **DROP TABLE Radnik SET brod = 10 WHERE brod=40**

2

467. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

**SELECT Odeljenje.Imeod, Radnik.Ime  
FROM Odeljenje LEFT JOIN Radnik ON Radnik.Brod = Odeljenje.Brod**

3

Одредити шта се види као резултат датог упита. Заокружити број испред траженог одговора:

1. Називи свих одељења и имена радника у њима
2. Називи само одељења у којима нема радника
3. Називи само одељења у којима има радника са именима радника у њима
4. Називи одељења у којима има радника са именима радника у њима и називи одељења у којима нема радника

468. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	Naziv	Mesto
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

**SELECT Odeljenje.Brod, Odeljenje.Imeod, COUNT(\*)  
FROM Radnik INNER JOIN Odeljenje ON Radnik.Brod = Odeljenje.Brod  
GROUP BY Odeljenje.Brod, Odeljenje.Imeod**

3

Одредити приказ који је резултат датог упита. Заокружити број испред траженог одговора:

1. Бројева и назива свих одељења
2. Бројева и назива свих одељења са бројем радника у њима
3. Бројева и назива одељења у којима има радника са бројем радника у њима
4. Бројева и назива одељења у којима нема радника

469. Дат је упит:

```
SELECT *
FROM ucenici
WHERE odeljenje=4 OR odeljenje=7 OR odeljenje=10
```

Заокружити оператор који би требало користити у датом упиту да би избегли вишеструко коришћење оператора OR:

1

1. LIKE
2. BETWEEN
3. AND
4. IN

470. Заокружити број испред исказа о подупитима који је тачан:

1. Подупит се пише под наводницима.
2. Подупит не сме да садржи групне функције.
3. Подупит се често користи у WHERE клаузули да врати вредности које испуњавају одређени услов.
4. Подупит се најчешће извршава последњи, након што се изврши главни (спољни) упит.

2

471. Табела **Zaposleni** садржи поља: Zaposleni\_Id, Ime, Prezime, Plata, Odsek\_Id.

Дат је упит:

```
SELECT Zaposleni_Id, Ime
FROM Zaposleni
WHERE Plata=(SELECT MAX(Plata) FROM Zaposleni GROUP BY Odsek_Id)
```

Одредити разлог због кога дати упит **НЕ** ради. Заокружити број испред траженог одговора:

2

1. Подупит се пише под наводницима.
2. Подупит не сме користити групне функције.
3. Подупит враћа више од једне врсте, а коришћен је оператор за поређење са једном вредношћу.
4. Подупит може користити само колоне које се налазе у спољњем упиту иза речи SELECT.

472. Табела **Zaposleni** садржи поља: Zaposleni\_Id, Ime, Prezime, Plata, Odsek\_Id.

Потребно је приказати ид одсека, најмању и највећу плату у сваком одсеку. Заокружити број испред упита који одговара претходном захтеву:

2

1. `SELECT Odsek_Id, MIN(Plata), MAX(Plata)
FROM Zaposleni HAVING Odsek_Id`
2. `SELECT Odsek_Id, MIN(Plata), MAX(Plata)
FROM Zaposleni GROUP BY Odsek_Id`
3. `SELECT Odsek_Id, MIN(Plata), MAX(Plata)
FROM Zaposleni
WHERE Odsek=Odsek_Id`
4. `SELECT MIN(Plata), MAX(Plata)
FROM Zaposleni GROUP BY Odsek_Id`

473.	<p>Заокружити број испред тачног одговора. Креiran је упит са подупитом, а испред подупита је постављен оператор једнакости (=). Упит ће функционисати уколико је број вредности које подупит враћа.</p> <ol style="list-style-type: none"> <li>1. Неограничен</li> <li>2. Само једна</li> <li>3. До две</li> <li>4. До пет</li> </ol>	2
474.	<p>Заокруживањем редног броја испред понуђеног одговора, изабрати исказ који најбоље описује природно спајање табела.</p> <ol style="list-style-type: none"> <li>1. Спој између две табеле које имају колоне са истим именом, типом података и дужином.</li> <li>2. Спој између две табеле који као резултат даје Декартов производ.</li> <li>3. Спој између две табеле код којих не постоје поља која имају исто име, тип и дужину.</li> <li>4. Спој који користи само једну табелу.</li> </ol>	2
475.	<p>Заокруживањем редног броја обележити клаузулу коју је потребно користи уколико листа иза резервисане речи SELECT садржи колону и агрегатну функцију:</p> <ol style="list-style-type: none"> <li>1. HAVING клаузулу</li> <li>2. GROUP BY клаузулу</li> <li>3. JOIN клаузулу</li> <li>4. ORDER BY клаузулу</li> </ol>	1
476.	<p>Табела <b>Zaposleni</b> садржи поља: <i>Zaposleni_Id</i>, <i>Ime</i>, <i>Prezime</i>, <i>Plata</i>, <i>Odsek_Id</i> и незавршени упит:</p> <pre style="margin-left: 40px;"><code>SELECT COUNT(Odsek_Id), Ime, Prezime FROM Zaposleni GROUP BY ?????</code></pre> <p>Одредити чиме је потребно заменити ????? да би упит био исправан. Заокружити број испред траженог одговора:</p> <ol style="list-style-type: none"> <li>1. <i>Odsek_Id</i></li> <li>2. <i>Odsek_Id, Ime</i></li> <li>3. <i>Odsek_Id, Ime, Prezime</i></li> <li>4. <i>Ime, Prezime</i></li> </ol>	2
477.	<p>Извршава се следећа SELECT наредба:</p> <pre style="margin-left: 40px;"><code>SELECT MAX(Plata), Odsek_Id FROM Zaposleni GROUP BY Odsek_Id</code></pre> <p>Заокруживањем броја испред одговарајућег исказа, означити које ће вредности бити приказане:</p> <ol style="list-style-type: none"> <li>1. Највећа плата коју има запослени у фирмама.</li> <li>2. Највећа плата у сваком одсеку.</li> <li>3. Запослени који има највећу плату.</li> <li>4. Запослени који имају највећу плату у сваком одсеку.</li> </ol>	2
478.	<p>Означити оператор поређења који захтева листу вредности. Заокружити тражени одговор:</p> <ol style="list-style-type: none"> <li>1. IN</li> <li>2. LIKE</li> <li>3. BETWEEN ... IN ...</li> <li>4. IS NULL</li> </ol>	1

479.	Заокружити број испред назива споја који је потребно користити да би приказали сви редови из табела <b>Zaposleni</b> и <b>Istoriya_zaposlenih</b> .	
	1. Унутрашњи спој 2. Десни спољашњи спој 3. Леви спољашњи спој 4. Потпуни спољашњи спој	2
480.	Извршава се следећа SELECT наредба:	
	<pre>SELECT MIN(Datum_Zaposlenja), Odsek_Id FROM Zaposleni GROUP BY Odsek_Id</pre>	
	Заокруживањем броја испред одговарајућег исказа, одредити које ће вредности бити приказане:	2
	1. Најранији датум запослења за сваки одсек предузећа. 2. Најранији датум запослења у целој табели ZAPOSLENI. 3. Датум запослења последњег запосленог радника у целом предузећу. 4. Датум запослења најстаријег запосленог радника у целом предузећу. 5. Датум запослења најстаријег запосленог радника у сваком одсеку предузећа.	
481.	Заокруживањем броја испред понуђеног одговора, означити функцију која враћа датум и време на рачунару који је сервер базе података.	
	1. DATE 2. DATETIME 3. SYSDATE 4. CURRENTDATE	1
482.	Заокруживањем броја испред понуђеног одговора, означити шта се дешава када се креира Декартов производ:	
	1. Табела се спаја са собом, свака колона са сваком колоном. 2. Табела се спаја са другом једнаком табелом. 3. Све врсте једне табеле се спајају са свим врстама друге табеле. 4. Све врсте које се не слажу у WHERE клаузули се приказују.	2
483.	Одредити тачан исказ о оператору ANY који се примењује са подупитом који враћа више вредности:	
	1. Оператор ANY може да се користи са резервисаном речју DISTINCT. 2. Оператор ANY је синоним за оператор ALL. 3. Оператор ANY упоређује са свим вредностима које враћа подупит. 4. Оператор ANY може да се користи са операторима LIKE и IN.	2
484.	Заокружити број под којим су наведени оператори поређења који се могу користити у упиту са подупитом, у случају када подупит враћа више врста:	
	1. IN, ANY и ALL 2. LIKE 3. BETWEEN... AND ... 4. =, < и >	2

485. Извршава се SELECT упит који у себи има подупит:

```
SELECT Prezime, Ime
FROM Ucenici
WHERE Odeljenje_Id IN
(SELECT Odeljenje_Id
FROM Podela
WHERE Nastavnik_Id=20)
```

2

Одредити реченицу о датом подупиту која је тачна:

1. Спољни упит се извршава пре угњеженог подупита.
2. Резултати унутрашњег упита се враћају спољном упиту.
3. Јавља се грешка ако и унутрашњи и спољни упит не враћају вредности.
4. Оба упита (и спољашњи и унутрашњи) морају вратити вредност да не би дошло до грешке.

486. Одредити реченицу која најбоље описује значење ANY оператора:

1. Једнак било ком члану листе.
2. Упоређује вредност са сваком вредношћу коју враћа подупит.
3. Упоређује вредност са првом вредностим коју враћа подупит.
4. Једнак свакој вредности у листи.

2

487. Потребно је креирати извештај који приказује имена свих производа чија је цена већа од просечне цене свих производа.

Заокружити број испред упита који одговара постављеном задатку:

1. 

```
SELECT naziv
FROM proizvod
WHERE cena > (SELECT AVG(cena) FROM proizvod)
```
2. 

```
SELECT naziv
FROM proizvod
WHERE cena > AVG(cena)
```
3. 

```
SELECT AVG(cena), naziv
FROM proizvod
WHERE cena > AVG(cena)
GROUP BY naziv
```
4. 

```
SELECT naziv
FROM (SELECT AVG(cena) FROM proizvod)
WHERE cena > AVG(cena)
```

3

488. Табела **ARTIKLI** садржи следеће колоне: artikl\_id, naziv, kategorija, cena, kolicina. Потребно је да се прикаже категорија и минимална цена артикла у свакој категорији.

Изабрати упит који даје тражени извештај:

1. 

```
SELECT kategorija, MIN(cena)
FROM artikli
GROUP BY cena
```
2. 

```
SELECT kategorija, MIN(cena)
FROM artikli
GROUP BY kategorija
```
3. 

```
SELECT MIN(cena), kategorija
FROM artikli
GROUP BY MIN(cena), kategorija
```
4. 

```
SELECT cena, MIN(kategorija)
FROM artikli
GROUP BY kategorija
```

2

489. Табела **RADIONICA** садржи следеће колоне: radinica\_id, naziv, zanat, lokacija\_id. Потребно је да се прикаже број радионица са јединственим занатом на свакој локацији.
- Изабрати упит који даје тражени извештај:
1. `SELECT location_id, COUNT(DISTINCT zanat)  
FROM radionica`
  2. `SELECT location_id, COUNT(zanat)  
FROM radionica  
GROUP BY lokacija_id`
  3. `SELECT location_id, COUNT(DISTINCT zanat)  
FROM radionica  
GROUP BY lokacija_id`
  4. `SELECT location_id, COUNT(DISTINCT zanat)  
FROM radionica  
GROUP BY zanat`
- 3
- 
490. Табела **RADNIK** садржи следеће колоне: radnik\_id, ime, prezime, odsek\_id, plata. Потребно је да се прикаже просечна зарада запослених у сваком одсеку.
- Изабрати упит који даје тражени извештај:
1. `SELECT plata(AVG)  
FROM radnik  
GROUP BY odsek_id`
  2. `SELECT AVG plata  
FROM radnik  
GROUP BY odsek_id`
  3. `SELECT AVG(plata)  
FROM radnik  
BY odsek_id`
  4. `SELECT AVG(plata)  
FROM radnik  
GROUP BY odsek_id`
- 2
- 
491. Одредити број спојева (**join**) који се морају применити за спајање пет табела:
1. 3 споја
  2. 4 споја
  3. 5 спојева
  4. Један више од броја табела
- 1
- 
492. Изабрати назив могућности SQL наредбе да враћа податке из две или више табела:
1. Селекција
  2. Пројекција
  3. Спајање
  4. Разврставање
- 1
- 
493. Уколико подупит формираног упита враћа већи број врста, заокружити број испред оператора који се не може користити у таквом упиту:
1. LIKE
  2. ANY
  3. ALL
  4. IN
- 2

494. Табела **PROIZVOD** садржи следеће колоне: proizvod\_id, opis, cena, lokacija\_id. Потребно је приказати оне производе који задовољавају следеће услове:
- **cena** сваког приказаног производа приказује се увећана за 10%.
  - **lokacija\_id** мора бити 2000, 2001 или 2002.
  - 10% оригиналне цене мора бити мање од 100 динара.
- Проценити који упит даје тражене податке:
1. 

```
SELECT cena, proizvod_id, cena*1.10
 FROM proizvod
 WHERE cena*0.10<100,00
 AND lokacija_id IN (2000, 2001, 2002)
```
  2. 

```
SELECT cena, proizvod_id, cena*0.10
 FROM proizvod
 WHERE cena*1.10>100,00
 AND (lokacija_id IN (2000, 2001, 2002))
```
  3. 

```
SELECT cena, proizvod_id, cena*1.10
 FROM proizvod
 WHERE cena*1.10<100,00
 AND lokacija_id = (2000, 2001, 2002)
```
  4. 

```
SELECT cena, proizvod_id, cena*1.10
 FROM proizvod
 WHERE cena*0.10<100,00
 AND (lokacija_id IN 2000, 2001, 2002)
```
495. Одредити шта ће се десити ако се користи DELETE наредба без WHERE клаузулe:
1. Све врсте у табели ће бити обрисане.
  2. Јавиће се грешка.
  3. Ни једна врста неће бити обрисана.
  4. Само ће једна врста бити обрисана.
496. Извршава се наредба:
- ```
DELETE * FROM ucenik
```
- Одредити шта ће се десити са табелом UCENIK. Заокружити број испред траженог одговора:
1. Биће обрисана табела UCENIK.
 2. Биће обрисана колона UCENIK
 3. Све врсте у табели UCENIK ће бити обрисане.
 4. Само ће прва врста у табели UCENIK бити обрисана.
497. Табела ZAPOSLENI има следећа поља (zaposleni_id, ime, prezime, plata, odsek_id). Потребно је повећати плату запосленима у IT сектору 12.5%. IT сектор има као шифру одсека – 15. Изабрати коју наредбу је потребно употребити да би се урадила тражена измена у табели:
1.

```
UPDATE zaposleni(plata)
      SET plata=plata*1,125
```
 2.

```
UPDATE zaposleni
      SET plata=plata*0.125
      WHERE odsek_id=15
```
 3.

```
UPDATE zaposleni
      SET plata=plata*1.125
      WHERE odsek_id=15
```
 4.

```
UPDATE zaposleni (plata)
      VALUES (plata*0.125)
      WHERE odsek_id=15
```

498. Табела **ZAPOSLENI** има следећа поља (zaposeleni_id, ime, prezime, plata, odsek_id, bonus). Потребно је доделити бонус свим запосленим у сектору 10, Вредност бонуса поставити на исти бонус који је додељен запосленом чији је id 1221:

1. `UPDATE zaposleni
SET bonus = SELECT bonus FROM zaposleni
WHERE zaposeleni_id = 1221`
2. `UPDATE zaposleni
SET bonus = (SELECT bonus FROM zaposleni
WHERE zaposeleni_id = 1221)`
3. `UPDATE zaposleni
SET bonus = (SELECT bonus FROM zaposleni
WHERE zaposeleni_id = 1221)
WHERE odsek_id = 10`
4. `UPDATE zaposleni
SET bonus = (SELECT bonus FROM zaposleni
WHERE zaposeleni_id = 1221
WHERE odsek_id = 10)`

3

499. Извршава се наредба:

```
SELECT COUNT(*)
FROM ucenik
```

Заокружити број испред исказа који описује шта ова наредба ради:

1. Јавиће се грешка јер нема WHERE клаузуле.
2. Биће приказан број врста у табели UCENIK.
3. Биће приказани сви подаци о ученицима из табеле а након тога приказаће се и њихов број.
4. Биће приказани сви подаци о ученицима.

2

500. Извршава се упит:

```
SELECT prezime, ime, email
FROM ucenik
ORDER BY prezime
WHERE prosek>=4.50
```

Наредба се неће извршити. Заокружити број испред разлога услед кога се наредба неће извршити:

2

1. Наредба се неће извршити једино ако нема ни једног одличног ученика.
2. Услов није добро написан.
3. Потребно је убацити HAVING клаузулу.
4. Потребно је променити редослед клаузула.

501. Извршава се следећа наредба:

```
SELECT ucenik_id, ime, prosek
FROM ucenik
ORDER BY ime, prosek
```

Изабрати исказ који описује шта се дешава када се наредба изврши:

1

1. Списак ученика се сортира нумерички.
2. Списак ученика се сортира алфабетски.
3. Списак ученика се сортира најпре нумерички, а онда алфабетски.
4. Списак ученика се сортира најпре алфабетски, а онда нумерички.

| | | |
|------|--|---|
| 502. | Означити исказ који одређује где се пише ORDER BY клаузула: | |
| | 1. Увек на крају упита.
2. У листи иза SELECT наредбе.
3. Испред WHERE клаузуле. | 1 |
| 503. | Обележити истинити исказ о властитом споју (self join): | |
| | 1. Алијаси табела се не могу користити за одређивање имена табеле.
2. Властити спој (self join) се може користити само уз помоћ погледа (view).
3. Алијаси табела се морају користити да би одредили имена табеле. | 1 |
| 504. | Изабрати исказ о GROUP BY клаузули који је истинит: | |
| | 1. Груписање се најпре врши по првој колони у листи која се налази иза резервисаних речи GROUP BY.
2. Груписање се најпре врши по последњој колони у листи која се налази иза резервисаних речи GROUP BY.
3. GROUP BY клаузула се не може користи са ORDER BY клаузулом. | 1 |
| 505. | Уколико је потребно табелу спојити саму са собом тада је потребно користити: | |
| | 1. Алијасе табела.
2. Алијасе колона.
3. HAVING клаузулу.
4. GROUP BY клаузулу. | 1 |
| 506. | Изабрати наредбу која се користи да се уклони поглед (view): | |
| | 1. DROP
2. DELETE
3. DROP VIEW
4. REMOVE VIEW | 1 |
| 507. | Уколико се кроз поглед (view) жели да се промене подаци он <u>НЕ СМЕ</u> садржати: | |
| | 1. WHERE клаузулу
2. Спој
3. Алијас колоне
4. GROUP BY клаузулу | 2 |
| 508. | Дат је упит за креирање погледа и наведени искази који се односе на дати упит. Заокружити број испред тачног исказа: | |
| | <pre>CREATE VIEW Pregled_Proseka AS SELECT UcenikID, Ime, Prezime, AVG(Ocena) AS Prosek FROM Testovi WHERE OdeljenjeID IN (1, 2, 3, 4) GROUP BY UcenikID, Ime, Prezime</pre> <ol style="list-style-type: none"> Подаци у табели Testovi се могу модификовати коришћењем погледа Pregled_Proseka Коришћењем датог погледа, подаци се могу само у додавати у табелу Testovi, али не и мењати Овако дат упит изазива грешку при извршењу Коришћењем датог погледа, подаци из табеле Testovi се могу само прегледавати, али не и додавати или мењати | 3 |

| | | |
|------|--|---|
| 509. | Корисник базе треба да креира извештаје који садрже имена, презимена и стручну спрему запослених. При томе корисник нема потребе да „види“ остале податке о запосленима. Заокружити број испред описа активности коју администратор базе прво предузима да би се реализовало наведено ограничење приступа: | 2 |
| | <ol style="list-style-type: none"> 1. Креира извештај за корисника (report) 2. Креира поглед (view) 3. Креира подупит (subquery) 4. Креира индекс (index) | |
| 510. | Означити како се назива спајање свих врста једне табеле са свим врстама друге табеле: | 1 |
| | <ol style="list-style-type: none"> 1. упаривање 2. производ 3. Декартов производ 4. Декартово спајање | |
| 511. | Означити тачан исказ: | |
| | <ol style="list-style-type: none"> 1. Окидачи (тригери) се углавном користе за контролу извршавања наредби SQL-а. 2. Окидачи (тригери) се углавном користе за чување базе података од вируса. 3. Окидачи (тригери) се углавном користе за очување интегритета података у бази података. | 2 |
| 512. | Изабрати тврђење које је истинито у вези ADO.NET-а: | |
| | <ol style="list-style-type: none"> 1. ADO.NET технологија представља модел састављен од скупа класа за приступ и манипулацију подацима који су смештени у различитим базама података. 2. ADO.NET је језик намењен приступу подацима који су смештени у различитим базама података. 3. ADO.NET технологија је скуп функција намењених коришћењу података на Интернету који су смештени у различитим базама података. | 1 |

У следећим задацима заокружите бројеве испред тражених одговора

| | | |
|------|---|---|
| 513. | Међу понуђеним алатима, обележити Case алате: | 3 |
| | <ol style="list-style-type: none"> 1. Rational Rose 2. Oracle Designer 3. .NET 4. Microsoft Visio 5. Java 6. SQL Express | |
| 514. | Обележити команде које се сматрају командама ажурирања података у бази података: | |
| | <ol style="list-style-type: none"> 1. Организовање податка 2. Додавање нових података 3. Брисање старих података 4. Враћање оштећених података у коректно стање 5. Измена постојећих података 6. Додела права приступа подацима 7. Измена структуре постојећих табела у бази | 3 |

| | | |
|------|---|---|
| 515. | Обележити компоненте које, поред хардвера, софтвера и рачунарских мрежа, чине информационо систем: | |
| | 1. подаци
2. језик за обраду података
3. базна компонента
4. мрежна компонента | 2 |
| 516. | Обележити делове манипулативне компоненте релационог система: | |
| | 1. Релациона алгебра
2. Релационе функције
3. Релациони рачун
4. Релациони типови | 2 |
| 517. | Одредити ентитете који садрже одговарајуће атрибуте: | |
| | 1. Ентитет: КЊИГА – Атрибути: наслов, аутор, издавач, година издања
2. Ентитет: АУТОМОБИЛ – Атрибути: марка, година производње, боја, власник, година рођења власника, регистарски број
3. Ентитет: УЧЕНИК – Атрибути: име, презиме, разред, одељење, број оправданих, број неоправданих, просек
4. Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина, главни град, број становника главног града, име градоначелника главног града | 2 |
| 518. | Одредити ентитете који садрже атрибуте који НИСУ одговарајући: | |
| | 1. Ентитет: КЊИГА – Атрибути: наслов, аутор, издавач, година издања
2. Ентитет: АУТОМОБИЛ – Атрибути: марка, година производње, боја, власник, година рођења власника, регистарски број
3. Ентитет: УЧЕНИК – Атрибути: име, презиме, разред, одељење, број оправданих, број неоправданих, просек
4. Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина, главни град, број становника главног града, име градоначелника главног града
5. ЕНТИТЕТ: САЈАМ – Атрибути: назив, датум почетка, датум завршетка, број излагача, број посетилаца | 2 |
| 519. | Одредити ентитете који садрже одговарајуће атрибуте: | |
| | 1. Ентитет: СТУДЕНТ – Атрибути: име, презиме, смер, број бодова, просек
2. Ентитет: КЊИГА – Атрибути: наслов, аутор, година издања, издавач, адреса издавача, телефон издавача
3. Ентитет: АВИОН – Атрибути: произвођач, марка, година производње, број седишта
4. Ентитет: ДРЖАВА – Атрибути: назив, број становника, површина
5. Ентитет: САЈАМ – Атрибути: назив, број излагача, покровитељ, адреса покровитеља, контакт особа покровитеља
6. Ентитет: ТУРИСТИЧКА АГЕНЦИЈА – Атрибути: назив, адреса, година оснивања, власник, стручна квалификација власника, запослени, стручна квалификација запослених | 3 |
| 520. | Одабрати све могуће инстанце ентитета ОСОБА: | |
| | 1. Новак Ђоковић
2. име и презиме
3. Никола Тесла
4. Женско
5. датум рођења | 1 |

| | | |
|------|---|---|
| 521. | Дати су искази који се односе на спољашњи кључ табеле (foreign key constraint). Заокружити бројеве испред тачних исказа: | |
| | 1. Вредност у пољу спољашњег кључа не сме бити NULL | |
| | 2. Вредност спољашњег кључа мора бити јединствена (unique) у колони над којом је постављено ограничење спољашњег кључа | 2 |
| | 3. Вредност у пољу спољашњег кључа мора бити или NULL или једнака некој од вредности из колоне на коју спољашњи кључ референцира | |
| | 4. Више редова у табели може садржати исту вредност у пољу спољашњег кључа и тиме показивати на исти ред у референцираној табели | |
| | 5. Колона спољашњег кључа не мора садржати исти тип података као колона на коју спољашњи кључ референцира | |
| 522. | Обележити ентитете код којих је извршен адекватан избор јединственог идентификатора ентитета: | |
| | 1. јединствени матични број грађанина (ЈМБГ) за ентитет ОСОБА | |
| | 2. датум рођења за ентитет ОСОБА | 3 |
| | 3. ИСБН број за ентитет КЊИГА | |
| | 4. регистарска ознака за АУТОМОБИЛ | |
| | 5. дестинација за ентитет АРАНЖМАН | |
| | 6. директор за ентитет ШКОЛА | |
| 523. | Заокружити бројеве испред наредби које служе за креирање, брисање и измену структуре релационе базе и објекта који чине релациону базу: | |
| | 1. ALTER TABLE | |
| | 2. INSERT | 3 |
| | 3. CREATE TABLE | |
| | 4. DROP TABLE | |
| | 5. UPDATE | |
| | 6. DELETE TABLE | |
| | 7. ADD COLUMN | |
| | 8. ADD CONSTRAINT | |
| 524. | За измену – ажурирање структуре табеле која се налази у оквиру базе података користи се: | |
| | 1. ALTER – ADD – за додавање колоне у табелу | |
| | 2. ALTER – DROP – за брисање – уклањање колоне из табеле | 2 |
| | 3. ALTER – INSERT – за додавање колоне у табелу | |
| | 4. ALTER – DELETE – за брисање – уклањање колоне из табеле | |
| 525. | Обележити клаузуле SQL наредбе у којима се могу користити аритметичке операције: | |
| | 1. SELECT | |
| | 2. FROM | 2 |
| | 3. WHERE | |
| | 4. ORDER BY | |
| 526. | Заокружити бројеве испред тражених одговора. Означити операторе поређења који се <u>не могу</u> користити у упиту са подупитом који враћа више вредности: | |
| | 1. ALL | |
| | 2. ANY | 2 |
| | 3. = | |
| | 4. > | |

527. За упите са специфицираним редоследом приказа врста у резултујућој табели користи се клаузула ORDER BY после које се наводи назив колоне:
1. и службена реч ASC за растући редослед
 2. и службена реч DESC за опадајући редослед
 3. и службена реч ASC за опадајући редослед
 4. и службена реч DESC за растући редослед
 5. службена реч се може изоставити, добија се растући поредак
 6. службена реч се може изоставити, добија се опадајући поредак
- 1,5
-
528. Поребно је у табелу под именом RADNIK додати примарни кључ - прогласити примарним кључем колону IDBR.
1. `ALTER TABLE RADNIK ADD CONSTRAINT PK_RADNIK PRIMARY KEY (IDBR);`
 2. `ALTER TABLE RADNIK ADD PRIMARY KEY (IDBR);`
 3. `CREATE TABLE RADNIK ADD PK_RADNIK PRIMARY KEY (IDBR);`
 4. `DROP TABLE RADNIK ADD PK_RADNIK PRIMARY KEY (IDBR);`
- 2
-
529. Дата је табела **RADNIK**:
- | IDBR | IME | PREZIME | PLATA | PREMIJA | DATZAP |
|------|----------|------------|-------|---------|------------|
| 5900 | Slobodan | Golubović | 900 | 1300 | 3.10.1978 |
| 5932 | Mitar | Gavrilović | 2600 | NULL | 25.3.1985 |
| 5953 | Persida | Kosanović | 1100 | 0 | 12.1.1979 |
| 6234 | Marko | Pavlović | 1300 | 3000 | 17.12.1990 |
| 6789 | Janko | Nikolić | 3900 | 10 | 31.12.1995 |
- Заокружити бројеве испред упита којим ће се приказати само презиме и име за раднике запослене у периоду од 1. јануара 1980. до 31. децембра 1995.
- 2
1. `SELECT Prezime, Ime FROM Radnik WHERE DatZap > '1980-01-01' AND < '1995-12-31'`
 2. `SELECT Prezime, Ime FROM Radnik WHERE DatZap >= '1980-01-01' AND DatZap <= '1995-12-31'`
 3. `SELECT Prezime, Ime FROM Radnik WHERE DatZap BETWEEN '1980-01-01' AND '1995-12-31'`
 4. `SELECT Prezime, Ime FROM Radnik WHERE DatZap BETWEEN ('1980-01-01', '1995-12-31')`
-
530. Дати су искази који се односе на атрибуте неког ентитета у бази података. Заокружити бројеве испред тачних исказа:
1. Атрибут ентитета мора имати кључ.
 2. Атрибут ентитета мора имати кардиналност.
 3. Атрибут ентитета мора имати тип.
 4. Атрибут ентитета мора имати опционалност.
 5. Атрибут ентитета мора имати назив.
- 2
-
531. Одредити које кључне речи се могу користити када се креира поглед (view):
1. HAVING
 2. WHERE
 3. OPTION
 4. INTO
- 2

532. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

| IDBR | IME | PREZIME | PLATA | BROD |
|------|----------|------------|-------|------|
| 5900 | Slobodan | Golubović | 900 | 10 |
| 5932 | Mitar | Gavrilović | 600 | 10 |
| 5953 | Persida | Kosanović | 1100 | 20 |
| 6234 | Marko | Pavlović | 1300 | 30 |
| 6789 | Janko | Nikolić | 800 | 10 |

| BROD | Naziv | Mesto |
|------|-----------|----------|
| 50 | Skladišta | Zemun |
| 30 | Marketing | Vračar |
| 10 | Plasman | Surčin |
| 20 | Direkcija | Grocka |
| 40 | Nabavka | Barajevo |

Дат је упит:

```
UPDATE Radnik SET Radnik.Brod = 50
WHERE Radnik.Brod=(SELECT Brod FROM Odeljenje WHERE Odeljenje.Imeod='Plasman')
```

Одредити којим типовима упита припада дати упит. Заокружити бројеве испред тражених одговора:

1. Параметарски упит
2. Агрегатни упит
3. Угњеждени упит
4. Акциони упит

2

533. Извршава се следећи упит:

```
SELECT cena
FROM proizvod
WHERE cena IN (101,125,150,350)
AND (cena BETWEEN 125 AND 140 OR cena >150)
```

Одредити које две вредности може вратити ова наредба. Заокружити бројеве испред тражених вредности:

2

1. 101
2. 150
3. 125
4. 110
5. 350

534. Изабрати које две команде се могу користити за ажурирање постојећих података у бази:

1. DELETE
2. MERGE
3. SELECT
4. UPDATE

2

535. Означити исказе који **НИСУ** тачни. Окидачи (тригери) су специјалне ускладиштене процедуре које се покрећу у следећим ситуацијама:

1. у сврху генерисања виртуелних табела у бази података.
2. као одговор на извршење INSERT, UPDATE и DELETE команде над табелом
3. као одговор на извршење CREATE, ALTER и DROP команде над објектима у бази
4. као одговор на LOGON догађај
5. у сврху чувања базе података од вируса и спољних напада
6. за контролу извршавања наредби SQL-а

3

536. Изабрати главне компоненте ADO.NET технологије:

1. Access
2. .NET Framework data providers
3. DataSet
4. Office

2

Допуните следеће реченице и табеле

537. Допунити реченицу:

Систем базе података у ужем смислу састоји се из базе података и

1

538. Допунити реченицу:

Навести средство којим корисник остварује комуникацију са релационом базом података:

1

539. Допунити реченицу:

Системи за управљање базом података данас имају углавном _____ архитектуру.

1

540. Допунити реченицу.

Навести компоненте релационог модела: _____, _____ и
_____.

3

541. Допунити реченицу:

Релациони модел података представља базу података као скуп _____.

1

542. Допунити реченицу наводећи назив нормалне форме:

Уколико ни један атрибут релације није вишевредносни, нити композитни, тј. не може се разставити, кажемо да је релација у _____ нормалној форми.

1

543. Допунити реченицу наводећи назив нормалне форме:

Уколико сви атрибути релације који нису део кључа зависе од сваког атрибута који је део кључа кажемо да је релација у _____ нормалној форми.

1

544. Допунити реченицу наводећи назив нормалне форме:

Уколико сви некључни (споредни) атрибути релације не зависе од неког другог некључног атрибута, тј. ако не постоји транзитивна зависност било ког споредног атрибута од било ког кључа те релације, кажемо да је релација у _____ нормалној форми.

1

545. Допунити реченицу:

Уколико је ентитет егзистенцијално зависан од неког ентитета са којим је у вези онда њега називамо _____ ентитет.

1

546. Допунити реченицу:

Поглед је _____ (изведена) табела, чији се садржај генерише на захтев _____.

2

547. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

| IDBR | IME | PREZIME | PLATA | BROD |
|------|----------|------------|-------|------|
| 5900 | Slobodan | Golubović | 900 | 10 |
| 5932 | Mitar | Gavrilović | 600 | 10 |
| 5953 | Persida | Kosanović | 1100 | 20 |
| 6234 | Marko | Pavlović | 1300 | 30 |
| 6789 | Janko | Nikolić | 800 | 10 |

| BROD | Naziv | Mesto |
|------|-----------|----------|
| 50 | Skladišta | Zemun |
| 30 | Marketing | Vračar |
| 10 | Plasman | Surčin |
| 20 | Direkcija | Grocka |
| 40 | Nabavka | Barajevo |

2

Извршава се упит:

```
UPDATE Radnik SET Radnik.Brod = 50
```

```
WHERE Radnik.Brod=(SELECT Brod FROM Odeljenje WHERE Odeljenje.Imeod='Plasman')
```

Допуните реченицу:

Извршењем упита ажурирају се подаци у записима за _____ радника.

548. Уписати редни број почев од 1 на линију испред резервисане речи тако да одговара редоследу навођења.

За формирање упита за издавање дела података из табеле која се налази у оквиру базе података користе се у следећем редоследу:

_____ GROUP BY
 _____ WHERE
 _____ SELECT
 _____ ORDER BY
 _____ FROM

3

У следећим задацима уредите и повежите појмове према захтеву

549. Исписати на цртама испред релација редни број под којим је наведена одговарајућа кардиналност везе:

- | | | |
|----|-------|-------------------------------|
| 1. | 1 : 1 | _____ ВЛАСНИК – БРОЈ ТЕЛЕФОНА |
| 2. | 1 : M | _____ НАСТАВНИК – ПРЕДМЕТ |
| 3. | M : M | _____ ОСОБА – ПАСОШ |

3

550. Написати на цртама испред логичких операција редне бројеве њихових приоритета:

- | | | |
|----|-------------------|-----------|
| 1. | највиши приоритет | _____ OR |
| 2. | средњи приоритет | _____ NOT |
| 3. | најнижи приоритет | _____ AND |

3

551. Повезати резервисану реч са редним бројем који одговара појму на који се односи у терминологији база података.

- | | | |
|----|-------------|-------------------|
| 1. | Табела | _____ VIEW |
| 2. | Страни кључ | _____ CONSTRAINT |
| 3. | Поглед | _____ TABLE |
| 4. | Окидач | _____ FOREIGN KEY |
| 5. | Ограниччење | _____ TRIGGER |

2,5

552. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

| IDBR | IME | PREZIME | PLATA | BROD |
|------|----------|------------|-------|------|
| 5900 | Slobodan | Golubović | 900 | 10 |
| 5932 | Mitar | Gavrilović | 600 | 10 |
| 5953 | Persida | Kosanović | 1100 | 20 |
| 6234 | Marko | Pavlović | 1300 | 30 |
| 6789 | Janko | Nikolić | 800 | 10 |

| BROD | Naziv | Mesto |
|------|-----------|----------|
| 50 | Skladišta | Zemun |
| 30 | Marketing | Vračar |
| 10 | Plasman | Surčin |
| 20 | Direkcija | Grocka |
| 40 | Nabavka | Barajevo |

Повезати упите и њихова значења уписом броја упита на одговарајућу линију:

1. `SELECT odeljenje.imeod,
radnik.prezime
FROM odeljenje INNER JOIN radnik
ON radnik.brod = odeljenje.brod`

Приказује све раднике, и оне који нису распоређени у одељења, и само она одељења у којима има радника

3

2. `SELECT odeljenje.imeod,
radnik.prezime
FROM odeljenje LEFT JOIN radnik
ON radnik.brod = odeljenje.brod`

Приказује само одељења у којима има радника и само раднике распоређене у одељењима

3. `SELECT odeljenje.imeod,
radnik.prezime
FROM odeljenje RIGHT JOIN radnik
ON radnik.brod = odeljenje.brod`

Приказује сва одељења, и она у којима нема радника и само оне раднике који су распоређени у одељења

553. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

| IDBR | IME | PREZIME | PLATA | BROD |
|------|----------|------------|-------|------|
| 5900 | Slobodan | Golubović | 900 | 10 |
| 5932 | Mitar | Gavrilović | 600 | 10 |
| 5953 | Persida | Kosanović | 1100 | 20 |
| 6234 | Marko | Pavlović | 1300 | 30 |
| 6789 | Janko | Nikolić | 800 | 10 |

| BROD | Naziv | Mesto |
|------|-----------|----------|
| 50 | Skladišta | Zemun |
| 30 | Marketing | Vračar |
| 10 | Plasman | Surčin |
| 20 | Direkcija | Grocka |
| 40 | Nabavka | Barajevo |

Повезати упите и њихова значења уписом броја датог испред описа значења упита на одговарајућу линију:

`SELECT odeljenje.imeod,
radnik.prezime
FROM odeljenje LEFT JOIN radnik
ON radnik.brod = odeljenje.brod
WHERE radnik.brod IS NULL`

1. Приказује само раднике који нису распоређени у одељења

3

`SELECT odeljenje.imeod,
radnik.prezime
FROM odeljenje FULL JOIN radnik
ON radnik.brod = odeljenje.brod`

2. Приказује све раднике, и оне који нису распоређени у одељења, и само она одељења у којима има радника

`SELECT odeljenje.imeod,
radnik.prezime
FROM odeljenje RIGHT JOIN
radnik
ON radnik.brod = odeljenje.brod
WHERE odeljenje.brod IS NULL`

3. Приказује сва одељења - и она у којима има и оне у којима нема радника и све раднике – и оне који су распоређени у одељења, као и оне који нису распоређени

4. Приказује само одељења у којима нема радника

АНЕКС 2

РАДНИ ЗАДАЦИ СА ОБРАСЦИМА ЗА ОЦЕЊИВАЊЕ НА МАТУРСКОМ ПРАКТИЧНОМ РАДУ

Поштовани ученици, ментори и оцењивачи,

Пред вама су радни задаци и обрасци за оцењивање који ће бити заступљени на матурском практичном раду за образовни профил **електротехничар информационих технологија - оглед**. Намењени су за вежбање и припрему за полагање матурског практичног рада, као и оцењивачима за усвајање примењене методологије оцењивања.

Задаци су распоређени према компетенцијама које се проверавају на испиту и то задаци са ознаком А односе се на компетенцију *Израда веб странице и веб апликације и објава на веб серверу*, док задаци означени словом Б одговарају компетенцији *Израда десктоп апликације, пројектовање једноставне базе података и повезивање апликације са базом података*. У оквиру сваког задатка проверава се ученикова компетентност и у погледу планирања и организације рада.

Група задатака А – *Израда веб странице и веб апликације и објава на веб серверу*

Група задатака Б – *Израда десктоп апликација, пројектовање једноставне базе података и повезивање апликације са базом података*

Задатком је предвиђено да се ученик „стави“ у професионалну ситуацију док извршава послове **електротехничара информационих технологија - оглед**.

Сваки радни задатак доноси **максимално 100 бодова**. Ученик мора остварити **најмање 50 бодова на сваком задатку** како би положио испит. Обрасци за оцењивање садрже утврђене аспекте, индикаторе оцењивања као и одговарајуће мере процене дате кроз двостепену скалу. Оцењивачи током оцењивања прате извршење задатка и вреднују сваки од индикатора за процену.

Радни задаци који ће бити реализовани на матурском испиту омогућавају проверу оспособљености ученика за обављање конкретних послова за квалификацију за коју су се школовали, као и утврђивање спремности за укључивање у свет рада.

Желимо вам срећан и успешан рад!

Аутори

КОМПЕТЕНЦИЈА А

ШИФРА ЗАДАТКА: ЕИТ-А1

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ДОМАЋЕ ЖИВОТИЊЕ

ЕИТ-А1: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ДОМАЋЕ ЖИВОТИЊЕ

Креирати веб апликацију са називом “Домаће животиње” која се састоји од три странице: Почетна, О аутору и Упутство.

На страници Почетна поставити позадину која треба да представља природно станиште домаћих животиња, а на њој поставити слике најмање пет домаћих животиња.

На страници О аутору написати основне податке о аутору.

На страници Упутство написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: Преласком миша преко слике домаће животиње чује се карактеристичан звук те животиње, а левим кликом миша на слику животиње отвара се прозор са кратким описом те домаће животиње.

CSS код и код за функционалност издвојити у посебне екстерне датотеке а учитати их у HTML при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А1

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А1 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ –ДОМАЋЕ ЖИВОТИЊЕ |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Креирање веб страница

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Креиране све веб странице према задатку | 5 | 0 |
| Странице имају одговарајући садржај | 6 | 0 |
| Функционишу све везе између страница | 8 | 0 |

2. Поставке почетне веб странице

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављена позадина на веб страницу | 5 | 0 |
| Постављене слике на веб страницу | 6 | 0 |
| Позадина и слике одговарају захтевима задатка | 8 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Десним кликом миша на слику отвара се нови прозор | 5 | 0 |
| Прозор има одговарајући садржај | 6 | 0 |
| Преласком миша преко слике чује се тражени звук | 8 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 24) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Слике и фајлови се налазе у посебним фолдерима | 5 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност је у посебном фајлу | 6 | 0 |
| Кодови за функционалност и CSS се учитавају при стартовању | 8 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Веб апликација постављена на Веб сервер | 5 | 0 |
| Апликација се покреће позивом фајла index.html | 6 | 0 |
| Сви линкови у апликацији правилно функционишу | 8 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А2

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ФОТО ГАЛЕРИЈА

ЕИТ-А2: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ФОТО ГАЛЕРИЈА

Креирати веб апликацију са називом “Фото галерија” која се састоји од три странице: Почетна, О аутору и Упутство.

На страници Почетна поставити контејнер за смештај галерије величине 640x480 px. На средини контејнера поставити централну фотографију величине 320x200px, а испод ње распоредити десет малих фотографија величине 64x40 px. Фотографије се учитавају из посебног фолдера и произвољних су димензија, а урађене су у JPEG формату. Фотографије имају називе slika1.jpg, slika2.jpg, i slika10.jpg.

На страници О аутору написати основне податке о аутору.

На страници Упутство написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: Левим кликом миша на доњу (малу) фотографију та фотографија се приказује на месту централне фотографије, а такође се исписује њен назив изнад централне фотографије.

CSS код и код за функционалност издвојити у посебне екстерне датотеке, а учитати их у HTML при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А2

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|---|
| Шифра радног задатка | ЕИТ-А2 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ФОТО ГАЛЕРИЈА |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|--|----|----|----|----|----|---------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Креирање веб страница

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Креиране све веб странице према задатку | 5 | 0 |
| Странице имају одговарајући садржај | 6 | 0 |
| Функционишу све везе између страница | 8 | 0 |

2. Поставке почетне веб странице

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављен контејнер на веб страницу | 5 | 0 |
| Постављена централна као и мале фотографије на веб страницу | 6 | 0 |
| Фотографије одговарају условима задатка | 8 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Десним кликом миша на малу фотографију мења се централна фотографија | 5 | 0 |
| Централна фотографија је адекватна малој фотографији на коју је кликнуто мишем | 6 | 0 |
| Изнад централне фотографије исписује се одговарајући назив | 8 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 24) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Фајлови са кодом и фотографијама се налазе у посебним фолдерима | 5 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност је у посебном фајлу | 6 | 0 |
| Кодови за функционалност и CSS се учитавају при стартовању | 8 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Веб апликација постављена на Веб сервер | 5 | 0 |
| Апликација се покреће позивом фајла index.html | 6 | 0 |
| Сви линкови у апликацији правилно функционишу | 8 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А3

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА

ЕИТ-А3: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА

Креирати веб апликацију са називом “Временска прогноза” која се састоји од три странице: Почетна, О аутору и Упутство.

На страници Почетна поставити форму за избор града чију временску прогнозу желимо да видимо. Испод форме поставити IFRAME link преузет са странице <http://www.naslovi.net> постављен за град Београд. Линк се такође може преузети и са неке друге странице која се бави временском прогнозом.

На страници О аутору написати основне податке о аутору.

На страници Упутство написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: На форми омогућити избор најмање за десет градова или локација у Србији. По избору локације, или града, у IFRAME приказати дневну временску прогнозу за дати избор.

CSS код и код за функционалност издвојити у посебне екстерне датотеке, а учитати их у HTML при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А3

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА / ЕИТ-АЗ

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-АЗ |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|----|----|----|----|----|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

КОМЕНТАРИ:

1. Креирање веб страница

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Креиране све веб странице према задатку | 5 | 0 |
| Странице имају одговарајући садржај | 6 | 0 |
| Функционишу све везе између страница | 8 | 0 |

2. Поставке почетне веб странице

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Постављена форма на страницу | 5 | 0 |
| Постављен IFRAME на страницу | 6 | 0 |
| Форма и IFRAME одговарају условима задатка | 8 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| На форми је могуће извршити избор града или локације | 5 | 0 |
| Подаци са форме се преносе у IFRAME | 6 | 0 |
| У IFRAME се приказује дневна прогноза за изабрану локацију | 8 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 24) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Фајлови са кодом и подацима се налазе у посебним фолдерима | 5 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност је у посебном фајлу | 6 | 0 |
| Кодови за функционалност и CSS се учитавају при стартовању | 8 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Веб апликација постављена на Веб сервер | 5 | 0 |
| Апликација се покреће позивом фајла index.html | 6 | 0 |
| Сви линкови у апликацији правилно функционишу | 8 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А4

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СРБИЈА И ЊЕНИ СУСЕДИ

ЕИТ-А4: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СРБИЈА И ЊЕНИ СУСЕДИ

Креирати веб апликацију са називом “Србија и њени суседи” која се састоји од три странице: Почетна, О аутору и Упутство.

На страници Почетна поставити позадину која треба да представља географску карту са суседима Републике Србије, при чему наша земља заузима централно место. На сваку од приказаних земаља на карти поставити њен назив, као и слику заставе те земље.

На страници О аутору написати основне податке о аутору.

На страници Упутство написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: Кликом миша на назив земље отвара се прозор у коме се приказују основни подаци о земљи (главни град, број становника, површина ...). Преласком миша преко заставе земље чује се неколико тактова химне те земље.

CSS код и код за функционалност издвојити у посебне екстерне датотеке, а учитати их у HTML при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А4

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А4 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СРБИЈА И ЊЕНИ СУСЕДИ |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

КОМЕНТАРИ:

1. Креирање веб страница

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Креиране све веб странице према задатку | 5 | 0 |
| Странице имају одговарајући садржај | 6 | 0 |
| Функционишу све везе између страница | 8 | 0 |

2. Поставке почетне веб странице

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављена позадина на веб страницу | 5 | 0 |
| Постављене слике на веб страницу | 6 | 0 |
| Позадина и слике одговарају захтевима задатка | 8 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Кликом миша на назив земље отвара се нови прозор | 5 | 0 |
| Отворени прозор има одговарајући садржај | 6 | 0 |
| Преласком миша преко слике заставе чује се тражени звук | 8 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 24) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Слике и фајлови се налазе у посебним фолдерима | 5 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност је у посебном фајлу | 6 | 0 |
| Кодови за функционалност и CSS се учитавају при стартовању | 8 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Веб апликација постављена на Веб сервер | 5 | 0 |
| Апликација се покреће позивом фајла index.html | 6 | 0 |
| Сви линкови у апликацији правилно функционишу | 8 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А5

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕЗЕРВАЦИЈА АУТОБУСКИХ КАРАТА

ЕИТ-А5: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕЗЕРВАЦИЈА АУТОБУСКИХ КАРАТА

Креирати веб апликацију са називом “Резервација аутобуских карата” која се састоји од три странице: Почетна, О аутору и Упутство.

На страници Почетна поставити приказ нумерисаних седишта аутобуса почев од броја 2 до 52, разврсаних у четири колоне. Испод приказа седишта поставити форму за слање мејла која треба да има следећа поља: Број седишта, Име и презиме, e-mail.

На страници О аутору написати основне податке о аутору.

На страници Упутство написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: При стартовању апликације слободна седишта су обележена зеленом бојом, а резервисана црвеном. Кликом миша на слободно седиште у поље форме Број седишта уноси се податак о седишту које желимо да резервишемо. Када се форма попуни осталим подацима и пошаље, слободно седиште мења боју у црвену, тј. постаје резервисано.

CSS код и код за функционалност издвојити у посебне екстерне датотеке, а учитати их у HTML при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А5

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А5 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕЗЕРВАЦИЈА АУТОБУСКИХ КАРАТА |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

КОМЕНТАРИ:

1. Креирање веб страница

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Креиране све веб странице према задатку | 5 | 0 |
| Странице имају одговарајући садржај | 6 | 0 |
| Функционишу све везе између страница | 8 | 0 |

2. Поставке почетне веб странице

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављен приказ седишта на веб страницу | 5 | 0 |
| Постављене форма за слање мејла на веб страницу | 6 | 0 |
| Приказ седишта и форма одговарају захтевима задатка | 8 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| При стартовању слободна седишта су обележена зеленом а резервисана црвеном бојом | 5 | 0 |
| Кликом миша на слободно седиште подаци се преносе на форму | 6 | 0 |
| После слања података са форме на мејл слободно седиште мења боју у црвену | 8 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 24) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Слике и фајлови се налазе у посебним фолдерима | 5 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност је у посебном фајлу | 6 | 0 |
| Кодови за функционалност и CSS се учитавају при стартовању | 8 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Веб апликација постављена на Веб сервер | 5 | 0 |
| Апликација се покреће позивом фајла index.html | 6 | 0 |
| Сви линкови у апликацији правилно функционишу | 8 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А6

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТЕЛЕФОНСКИ ИМЕНИК

ЕИТ-А6: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТЕЛЕФОНСКИ ИМЕНИК

Дата је текстуална датотека под називом „imenik.txt“ која представља колекцију онлајн података о претплатницима фиксне и мобилне телефоније.

Сваки ред у текстуалној датотеци представља податке о једном претплатничком броју у облику:

Шифра корисника | Име | Презиме | Адреса | Место | Број телефона | e-mail

Креирати веб апликацију под називом “Телефонски именик” која се састоји од три странице: Именик, Важни телефони и Корисничко упутство.

На страници „Именик“ поставити поља за унос параметара за претраживање телефонског именика, тастер „Тражи“ и линкове за прелаз на остале странице. Потребно је омогућити претраживање именика по следећим пољима:

Име – поље за унос текста са одговарајућом лабелом,

Презиме - поље за унос текста са одговарајућом лабелом,

Адреса - поље за унос текста са одговарајућом лабелом,

Место - падајућа листа са одговарајућом лабелом (DropDownList) са предефинисаним вредностима,

Број телефона - поље за унос текста са одговарајућом лабелом.

Код текстуалних поља омогућити претраживање тако да се унети узорак садржи у одговарајућем запису који се тражи (Ако корисник унесе у поље Име узорак „Мил“ апликација треба да излиста податке који у пољу Име садрже унети узорак нпр. **Милан, Милена, Милица, Милосав...**).

Поље за избор Места реализовано падајућом листом напунити свим различитим местима која се јављају у текстуалној датотеци са подацима.

Резултат претраге представити у облику табеле која садржи сва поља (колоне) која су дефинисана у датотеци са подацима.

На страници „Важни телефони“ приказати списак важних телефона: полиција (192), ватрогасци (193), хитна помоћ (194), тачно време (195), помоћ на путу (1987) и линкове за прелаз на остале странице.

На страници „Упутство“ написати кратко упутство за коришћење веб апликације и линкове за прелаз на остале странице.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Попуњена текстуална датотека која представља колекцију података за претраживање – телефонски именик

За оцењивање користити образац за оцењивање број ЕИТ-А6

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А6 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТЕЛЕФОНСКИ ИМЕНИК |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

КОМЕНТАРИ:

1. Поставке почетне странице

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Постављена сва текстуална поља | 6 | 0 |
| Постављено поље са падајућом листом (DropDownList) | 4 | 0 |
| Поставњен тастер „Тражи“ | 3 | 0 |
| Постављене одговарајуће лабеле поред поља за унос параметара претраге | 4 | 0 |
| Постављени линкови ка осталим страницама | 3 | 0 |

2. Поставке осталих страница

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|--|--------------|----------------|
| Постављена страница за преглед важних телефона | 4 | 0 |
| На страници за преглед важних телефона постављени линкови ка осталим страницама | 3 | 0 |
| Постављена страница за преглед корисничког упутства | 5 | 0 |
| На страници за преглед корисничког упутства постављени линкови ка осталим страницама | 3 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 30) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Резултати претраге се приказују у табели испод параметара за претрагу | 10 | 0 |
| У поље са падајућом листом су учитани одговарајући подаци | 5 | 0 |
| Функционише претрага по узорку код текстуалних поља | 8 | 0 |
| Функционише претрага код поља са падајућом листом | 7 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Подаци су меморијски представљени у структуре или класе | 9 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност и изглед је у посебном фајлу | 6 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Веб апликација постављена на Веб сервер | 10 | 0 |
| Сви ресурси апликације су доступни | 5 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А7

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕД ВОЖЊЕ

ЕИТ-А7: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕД ВОЖЊЕ

Дат је фолдер са текстуалним датотекама тако да се у свакој од датотека чувају подаци о реду вожње за једну аутобуску линију у оба смера.

Назив датотеке представља назив линије (нпр. Linija7.txt). Свака датотека садржи податке о поласцима у следећем облику:

SMER:Niš - Beograd

06:00

07:35

08:20

SMER:Beograd - Niš

06:05

07:40

08:50

09:30

Креирати веб апликацију под називом “Ред вожње” која се састоји од три странице: Ред вожње, Контакт и Корисничко упутство.

На страници “Ред вожње” поставити поља за избор параметара за претраживање и преглед реда вожње, тастер „Тражи“ и линкове за прелаз на остале странице. Претраживање реализовати по следећим пољима:

Аутобуска линија – падајућа листа са одговарајућом лабелом (DropDownList) са предефинисаним вредностима,

Смер - падајућа листа са одговарајућом лабелом (DropDownList) са предефинисаним вредностима

Поље за избор Аутобуске линије реализовано падајућом листом напунити свим различитим аутобуским линијама које су представљене фајловима у дефинисаном фолдеру у коме се чува ред вожње.

Поље за избор Смера реализовано падајућом листом динамички пунити смеровима који одговарају аутобуској линији која је изабрана у предходно дефинисаној падајућој листи „Аутобуске линије“.

Резултат претраге представити у облику табеле која садржи две колоне:

- Редни број поласка
- Време поласка

На страници „Контакт“ приказати текстуална поља за унос Имена, e-mail адресе, наслова поруке и поруке преко које корисници могу евентуално да пошаљу коментаре и питања у виду порука, као и линкове за прелаз на остале странице. Функционалност слања порука није потребно реализовати.

На страници „Упутство“ написати кратко упутство за коришћење веб апликације и линкове за прелаз на остале странице.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталиран на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Попуњене текстуалне датотеке које представљају линије са поласцима и смеровима аутобуског саобраћаја.

За оцењивање користити образац за оцењивање број ЕИТ-А7

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А7 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РЕД ВОЖЊЕ |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|----|----|----|----|----|----------------------|
| | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Поставке почетне странице

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Постављено поље са падајућом листом (DropDownList) Линије | 5 | 0 |
| Постављено поље са падајућом листом (DropDownList) Смерови | 5 | 0 |
| Поставњен тастер „Тражи“ | 3 | 0 |
| Постављене одговарајуће лабеле поред поља за унос параметара претраге | 4 | 0 |
| Постављени линкови ка осталим страницама | 3 | 0 |

2. Поставке осталих страница

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|--|--------------|----------------|
| Постављена страница за контакт | 4 | 0 |
| Постављена текстуална поља на страници за контакт | 5 | 0 |
| На страници за контакт постављени линкови ка осталим страницама | 3 | 0 |
| Постављена страница за преглед корисничког упутства | 5 | 0 |
| На страници за преглед корисничког упутства постављени линкови ка осталим страницама | 3 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 25) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Резултати претраге се приказују у табели испод параметара за претрагу | 7 | 0 |
| У поље са падајућом листом за линије су учитани одговарајући подаци | 5 | 0 |
| У поље са падајућом листом за смерове су учитани одговарајући подаци | 6 | 0 |
| Функционише претрага по линијама и смеровима | 7 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Подаци су меморијски представљени у структуре или класе | 9 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност и изглед је у посебном фајлу | 6 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Веб апликација постављена на Веб сервер | 10 | 0 |
| Сви ресурси апликације су доступни | 5 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А8

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВЕБ ПРОДАВНИЦА

ЕИТ-А8: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВЕБ ПРОДАВНИЦА

Дат је фолдер са slikama производа и текстуална датотека под називом „vebprodavnica.txt“ која представља колекцију података о артиклима веб продавнице.

Сваки ред у текстуалној датотеци представља податке о једном артиклу облику:

- **Шифра артика** – памти се као алфа нумерички податак у сваком реду текстуалне датотеке на позицији од индекса 0 до 5
- **Назив артика** - памти се као алфа нумерички податак у сваком реду текстуалне датотеке на позицији од индекса 6 до 30
- **Произвођач** - памти се као алфа нумерички податак у сваком реду текстуалне датотеке на позицији од индекса 31 до 50
- **RAM меморија** - памти се као алфа нумерички податак у сваком реду текстуалне датотеке на позицији од индекса 51 до 55
- **Тип процесора** - памти се као алфа нумерички податак у сваком реду текстуалне датотеке на позицији од индекса 56 до 70
- **Камера** - памти се као алфа нумерички податак у сваком реду текстуалне датотеке на позицији од индекса 71 до 80
- **Екран** - памти се као нумерички податак у сваком реду текстуалне датотеке на позицији од индекса 81 до 85
- **Слика** - памти се путања до слике у сваком реду текстуалне датотеке на позицији од индекса 91 до 120
- **Цена** - памти се као нумерички податак у сваком реду текстуалне датотеке на позицији од индекса 121 до 130

Креирати веб апликацију под називом “Веб продавница” која се састоји од две странице: Веб продавница и Корисничко упутство.

На страници „Веб продавница“ поставити поља за унос параметара за претраживање артикала веб продавнице, тастер „Тражи“ и линк за прелаз на страницу за упутство. Потребно је омогућити претраживање артикала по следећим пољима: **Произвођач, RAM меморија, Процесор, Камера, Екран**.

Сва поља за унос параметара претраге треба реализацијати као падајуће листе са одговарајућом лабелом (DropDownList) и предефинисаним вредностима (нпр.поље за избор произвођача напунити свим различитим произвођачима која се јављају у текстуалној датотеци са подацима, поље за избор RAM меморије напунити следећим вредностима: **512MB, 1GB, 1.5GB, 2GB и 3GB**).

Имплементирати механизам за претрагу који ће према задатом критеријуму приказати производе који одговарају параметрима претраге.

Резултат претраге представити у облику табеле која садржи сва поља (колоне) која су дефинисана у датотеци са подацима укључујући и слику производа приказану у првој колони.

На страници „Упутство“ написати кратко упутство за коришћење веб апликације и линк за прелаз на страницу веб продавнице.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације. Готову апликацију поставити на веб сервер инсталан на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Попуњена текстуална датотека која представља колекцију података о артиклима
- Фолдер са slikama производа

За оцењивање користити образац за оцењивање број ЕИТ-А8

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А8 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВЕБ ПРОДАВНИЦА |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

КОМЕНТАРИ:

1. Поставке почетне странице

| ИНДИКАТОРИ (максималан број бодова 25) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављена поља за унос параметара претраге (DropDownList) | 10 | 0 |
| Постављен тастер „Тражи“ | 3 | 0 |
| Постављене одговарајуће лабеле поред поља за унос параметара претраге | 4 | 0 |
| Постављен линк ка страници за упутство | 3 | 0 |
| Постављена табела за приказ резултата претраге | 5 | 0 |

2. Поставке осталих страница

| ИНДИКАТОРИ (максималан број бодова 10) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Постављена страница за преглед корисничког упутства | 7 | 0 |
| На страници за преглед корисничког упутства постављен линк ка страници за веб продавнику | 3 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 30) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Резултати претраге се приказују у табели | 10 | 0 |
| У поље са падајућом листом за произвођаче су учитани одговарајући подаци | 7 | 0 |
| Падајуће листе са фиксним вредностима са напуњене одговарајућим подацима | 7 | 0 |
| Слика производа је приказана у резултатима претраге | 6 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Подаци су меморијски представљени у структуре или класе | 9 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност и изглед је у посебном фајлу | 6 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Веб апликација постављена на Веб сервер | 10 | 0 |
| Сви ресурси апликације су доступни | 5 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А9

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КОНВЕРТОР ТЕКСТА

ЕИТ-А9: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КОНВЕРТОР ТЕКСТА

Написати XML веб сервис који има две јавне веб методе:

- string CyrilicaULatinicu (string tekst)
- string LatinicaUCirilicu (string tekst)

које врше превођење текста написаног ћирилицом у латиницу и превођење текста написаног латиницом у ћирилицу респективно.

Након тога креирати веб апликацију под називом “Конвертор текста” која се састоји од две странице: Конвертор и Упутство.

На страници „Конвертор“ поставити текстуална поља за текст написан латиницом и текст написан ћирилицом са одговарајућим лабелама, линк до упутства и тастере за иницирање акција за превођење из једног у друго писмо, и обрнуто.

Имплементирати механизам који ће активирањем одговарајућег тастера за превођење из једног у друго писмо извршити позив одговарајуће методе веб сервиса и резултат обраде приказати у одговарајуће поље за конвертовани текст.

На страници „Упутство“ написати кратко упутство за коришћење веб апликације и линк за прелаз на страницу Конвертор.

На страни веб сервиса логику за конверзију текста организовати у одговарајуће класе и структуре.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталан на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А9

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|---|
| Шифра радног задатка | ЕИТ-А9 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КОНВЕРТОР ТЕКСТА |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|--|----|----|----|----|----|---------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Поставке почетне странице

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|--|--------------|----------------|
| Постављена поља за унос и приказ преведеног текста и текста за превођење | 10 | 0 |
| Постављени тастери за иницирање акције превођења | 4 | 0 |
| Постављене одговарајуће лабеле поред текстуалних поља | 3 | 0 |
| Постављен линк ка страници за упутство | 3 | 0 |

2. Поставке осталих страница

| ИНДИКАТОРИ (максималан број бодова 8) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Постављена страница за преглед корисничког упутства | 5 | 0 |
| На страници за преглед корисничког упутства постављен линк ка страници за конвертор | 3 | 0 |

3. Функционалност веб апликације и веб сервиса

| ИНДИКАТОРИ (максималан број бодова 32) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Креирана веб метода за превођење ћирилице у латиницу | 8 | 0 |
| Креирана веб метода за превођење латинице у ћирилицу | 8 | 0 |
| Реализован позив веб сервиса на клијентској страни | 6 | 0 |
| Функционише превођење и приказ преведеног текста из латинице у ћирилицу | 5 | 0 |
| Функционише превођење и приказ преведеног текста из ћирилице у латиницу | 5 | 0 |

4. Организација података веб сервиса и апликације

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Подаци су меморијски представљени у структуре или класе | 10 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност и изглед је у посебном фајлу | 5 | 0 |

5. Поставке апликације и веб сервиса на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|--------------|----------------|
| Веб апликација постављена на Веб сервер | 10 | 0 |
| Веб сервис постављен не Веб сервер | 10 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А10

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КУРСНА ЛИСТА

ЕИТ-А10: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КУРСНА ЛИСТА

аписати XML веб сервис који има следеће јавне веб методе:

- double ProcitajKursNaDan(DateTime Datum, string valuta) – чита средњи курс на изабрани дан захтеване валуте из текстуалне датотеке
- bool UpisiKursNaDan(DateTime Datum, string valuta, double Kurs) – уписује средњи курс жељене валуте на одређени дан у текстуалну датотеку
- List<string> ProcitajSveValute() – враћа листу ознака свих различитих валута смештених у текстуалној датотеци

Структура датотеке у којој се чувају подаци је организована тако да се у једном реду датотеке памти курс једне валуте за један дан:

Датум | Ознака валуте | Курс

Након тога креирати веб апликацију под називом “Курсна листа” која се састоји од две странице:

- страница за преглед курса на дан.
- страница за ажурирање курсне листе.

На страници за преглед курса на дан поставити поље за избор датума (DatePicker), падајућу листу за избор валуте (DropDownList) коју треба напунити свим различитим валутама које враћа веб сервис, тастер за иницирање акције читања курса са веб сервиса и поље за приказ курса.

Имплементирати механизам који ће активирањем одговарајућег тастера, посредством веб сервиса, приказати вредност курса за изабрану валуту на изабрани дан или приказати поруку да не постоји тражени податак.

На страници за ажурирање курсне листе поставити поље за избор датума (DatePicker), падајућу листу за избор валуте (DropDownList) коју треба напунити свим различитим валутама које враћа веб сервис, поље за унос курса и тастер за иницирање акције уписа курса на веб сервис.

Имплементирати механизам који ће активирањем одговарајућег тастера извршити упис унетих података у текстуалну датотеку путем веб сервиса.

На страни веб сервиса логику организовати у одговарајуће класе и структуре.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталан на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А10

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А10 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КУРСНА ЛИСТА |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Поставке почетне странице

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављено поља избор датума (DatePicker) | 4 | 0 |
| Постављено поља за избор ознаке валуте (DropDownList) | 4 | 0 |
| Постављен тастер за иницирање акције одређивања курса | 2 | 0 |
| Постављене одговарајуће лабеле поред текстуалних поља | 2 | 0 |
| Постављено поље за приказ резултата обраде | 3 | 0 |

2. Поставке осталих странице

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављено поље избор датума (DatePicker) | 4 | 0 |
| Постављено поље за избор ознаке валуте (DropDownList) | 4 | 0 |
| Постављено поље за унос курса | 3 | 0 |
| Постављен тастер за иницирање акције снимања података | 2 | 0 |
| Постављене одговарајуће лабеле поред текстуалних поља | 2 | 0 |

3. Функционалност веб апликације и веб сервиса

| ИНДИКАТОРИ (максималан број бодова 30) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Креирана веб метода за читање курса из датотеке | 7 | 0 |
| Креирана веб метода за упис курса у датотеку | 7 | 0 |
| Креирана веб метода за читање свих различитих ознака валуте | 6 | 0 |
| Функционише читање и приказ курса на дан у предвиђено поље на веб страници | 5 | 0 |
| Функционише упис новог податка на креираној веб страници | 5 | 0 |

4. Организација података веб сервиса и апликације

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Подаци су меморијски представљени у структуре или класе | 10 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност и изглед је у посебном фајлу | 5 | 0 |

5. Поставке апликације и веб сервиса на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Веб апликација постављена на Веб сервер | 10 | 0 |
| Веб сервис постављен на Веб сервер | 10 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А11

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – НУМЕРОЛОШКО СЛАГАЊЕ

ЕИТ-А11: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – НУМЕРОЛОШКО СЛАГАЊЕ

Креирати веб апликацију, намењену за разоноду, са називом “Нумеролошко слагање”.

Посетити сајт <http://www.freewebtemplates.com/jquery-templates/> и са њега скинути неки темплејт по жељи. На основу тог темплејта креирати веб апликацију која треба да има најмање три странице: Почетна, О аутору и Упутство.

На страници Почетна поставити форму за унос имена и датума рођења за две особе:

Датум рођења особе1: Датум рођења особе2:

Испод форме поставити пригодну слику (нпр. два цвата, два лептира, два срца...) на левој страни слике треба да буде име особе1, на десној име особе 2, а у средини број који у процентима изражава колико се те две особе слажу.

На страници О аутору написати основне податке о аутору.

На страници Упутство написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: Попуњавањем комплетне форме израчунати проценат колико се две особе слажу. Тај проценат уписати у поље на средини слике. Проценат слагања рачунамо на следећи начин, сабирајмо цифре датума рођења све док не добијемо једноцифрен број, а то урадимо за оба датума рођења, потом поделимо мањи број са већим и помножимо са 100.

Пример израчунавања процената слагања за датуме 4.5.2001 и 9.10.2001:

$$4+5+2+0+0+1=12 \text{ на } 1+2=3$$

$$9+1+0+2+0+0+1=13 \text{ да } 1+3=4$$

Проценат слагања $(3/4) * 100 = 75\%$

CSS код и код за функционалност издвојити у посебне екстерне датотеке, а учитати их у HTML при стартовању апликације.

Готову апlicaцiю поставити на Веб сервер инсталлиран на локалном рачунару.

Прибор за рад;

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А11

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А11 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – НУМЕРОЛОШКО СЛАГАЊЕ |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Креирање веб страница

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављен одговарајући темплејт према задатку | 5 | 0 |
| Креиране веб странице са одговарајућим садржајима | 6 | 0 |
| Функционишу све везе између страница | 8 | 0 |

2. Поставке почетне веб странице

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Постављена форма на веб страницу | 5 | 0 |
| Постављене слике на веб страницу | 6 | 0 |
| Форме слике одговарају захтевима задатка | 8 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Подаци са форме се правилно прихватају | 5 | 0 |
| Правилно се израчунавају тражени подаци | 6 | 0 |
| Израчунати подаци се испisuју у одговарајућа поља | 8 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 24) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Слике и фајлови се налазе у посебним фолдерима | 5 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност је у посебном фајлу | 6 | 0 |
| Кодови за функционалност и CSS се учитавају при стартовању | 8 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Веб апликација постављена на Веб сервер | 5 | 0 |
| Апликација се покреће позивом фајла index.html | 6 | 0 |
| Сви линкови у апликацији правилно функционишу | 8 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А12

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КЊИГА УТИСАКА

ЕИТ-А12: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КЊИГА УТИСАКА

Креирати веб апликацију са називом “Књига утисака” која се састоји од три странице: Почетна, О аутору и Упутство.

На страници Почетна поставити форму за унос података која треба да има следећи изглед:

The form is titled "Knjiga utisaka". It has three required fields: "Ime" (Name), "Email", and "Komentar" (Comment). The "Komentar" field is a large text area with scroll bars. A button at the bottom says "Dodaj komentar" (Add comment).

На страници О аутору написати основне податке о аутору.

На страници Упутство написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: Сви подаци на форми морају бити унешени, а подаци за мејл адресу морају бити у исправном облику. Притиском на дугме `Dodajkomentar`, подаци са форме као и системски датум и време, се смештају у базу података. База података има само једну табелу, „Utisak”, која има следећа поља: ID, Ime, Email, Komentar и Datum.

CSS код и код за функционалност издвојити у посебне екстерне датотеке, а учитати их при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А12

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А12 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КЊИГА УТИСАКА |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Креирање веб страница

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Креиране све веб странице према задатку | 5 | 0 |
| Странице имају одговарајући садржај | 6 | 0 |
| Функционишу све везе између страница | 8 | 0 |

2. Поставке почетне веб странице

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављена форма на веб страницу | 5 | 0 |
| Форма има све елементе према задатку | 6 | 0 |
| Постављене контроле за валидацију форме | 8 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 24) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Све контроле на форми функционишу исправно | 5 | 0 |
| Креирана је база података са траженим елементима | 5 | 0 |
| Подаци са форме се прихватају и преносе у базу | 6 | 0 |
| Подаци су правилно уписаны у бази | 8 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност је у посебном фајлу | 6 | 0 |
| Кодови за функционалност и CSS се учитавају при стартовању | 8 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 19) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Веб апликација постављена на Веб сервер | 5 | 0 |
| Апликација се покреће позивом фајла index.html | 6 | 0 |
| Сви линкови у апликацији правилно функционишу | 8 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А13

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЦД КАТАЛОГ

ЕИТ-А13: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЦД КАТАЛОГ

Дата је текстуална датотека под називом „katalog.txt“ која представља колекцију података о албумима.

Сваки ред у текстуалној датотеци представља податке о једном албуму у облику:

Извођач | Назив албума | Жанр | Година издавања | Издавачка кућа|Слика омота

при чему је за податак „Слика омота“ дат путањом до локације где се слика чува на чврстом диску.

Креирати веб апликацију под називом “ЦД Каталог” која се састоји од две странице: „Каталог“ и „Корисничко упутство“.

На страници „Каталог“ поставити поља за унос параметара за претраживање података о албумима, тастер „Тражи“ и линк за прелаз на страницу за упутство. Потребно је омогућити претраживање каталога по следећим пољима:

Извођач – поље за унос текста са одговарајућом ознаком (Label),

Назив албума - поље за унос текста са одговарајућом ознаком (Label),

Жанр - падајућа листа (DropDownList) са одговарајућом ознаком (Label) са предефинисаним вредностима

Година издања - падајућа листа (DropDownList) са одговарајућом ознаком (Label) са предефинисаним вредностима

Издавачка кућа - поље за унос текста са одговарајућом ознаком (Label).

Код текстуалних поља омогућити претраживање тако да се унети узорак садржи у одговарајућем запису који се тражи. На пример, ако корисник унесе у поље Извођач узорак „Зор“, апликација треба да излиста податке који у пољу Извођач садрже унети узорак, нпр. **Зоран, Зорана, Зорица...**

Поље за избор године издања реализовано падајућом листом напунити подацима о свим различитим годинама издања који се јављају у текстуалној датотеци са подацима.

Поље за избор жанра реализовано падајућом листом напунити подацима о музичким жанровима (нпр. Поп, Рок, Етно, Фолк...).

Имплементирати механизам за претрагу који ће према задатом критеријуму приказати албуме који одговарају параметрима претраге.

Резултат претраге представити у облику табеле која садржи сва поља (колоне) која су дефинисана у датотеци са подацима, укључујући и слику омота албума коју треба приказати у првој колони.

На страници „Упутство“ написати кратко упутство за коришћење веб апликације и линк за прелаз на страницу „Каталога“.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Попуњена текстуална датотека која представља колекцију података о албумима
- Фолдер са slikama омота албума

За оцењивање користити образац за оцењивање број ЕИТ-А13

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А13 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЦД КАТАЛОГ |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|-----------|-----------|-----------|-----------|-----------|----------------------|
| Аспекти | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Поставке почетне странице

| ИНДИКАТОРИ (максималан број бодова 25) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављена сва текстуална поља | 6 | 0 |
| Постављена поља са падајућом листом (DropDownList) | 4 | 0 |
| Постављен тастер „Тражи“ | 3 | 0 |
| Постављене одговарајуће ознаке (Label) поред поља за унос параметара претраге | 4 | 0 |
| Постављен линк ка страници за упутство | 3 | 0 |
| Постављена табела за приказ резултата претраге | 5 | 0 |

2. Поставке осталих страница

| ИНДИКАТОРИ (максималан број бодова 8) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Постављена страница за преглед корисничког упутства | 5 | 0 |
| На страници за преглед корисничког упутства постављен линк ка страници „Каталог“ | 3 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 32) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| У поље са падајућом листом за годину издања су учитани одговарајући подаци из текстуалне датотеке | 6 | 0 |
| У поље са падајућом листом за жанр су учитани одговарајући предефинисани подаци | 5 | 0 |
| Функционише претрага по узорку код текстуалних поља | 8 | 0 |
| Функционише претрага код поља са падајућом листом | 7 | 0 |
| Слика албума је приказана у резултатима претраге | 6 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Подаци су меморијски представљени у структуре или класе | 9 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност и изглед је у посебном фајлу | 6 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Веб апликација постављена на Веб сервер | 10 | 0 |
| Сви ресурси апликације су доступни | 5 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А14

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТВ ПРОГРАМ

ЕИТ-А14: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТВ ПРОГРАМ

Дат је фолдер са текстуалним датотекама при чему се у свакој од датотека чувају подаци о дневној програмској шеми телевизијске станице.

Назив датотеке представља датум на који се програмска шема односи (нпр. 01.06.2015.txt). Свака датотека садржи податке о програмској шеми у следећем облику:

Време емитовања | Назив емисије | Тип емисије (текстуални облик)|

Тип емисије (графички приказ у виду иконе)

При чему је за податак о икони која представља графички приказ типа емисије дат путањом до локације где се она чува на чврстом диску.

Пример дела датотеке:

...
15:15 | Тесна кожа | Филм |C:\TV Program\Slake\Film.png
17:00 | Вести | Информативни програм |C:\TV Program\Slake\Informativni.png
17:15 | Тесна кожа 2 | Филм |C:\TV Program\Slake\Film.png

Креирати веб апликацију под називом “ТВ Програм” која се састоји од три странице: Програм, Контакт и Корисничко упутство.

На страници „Програм“ поставити поља за избор параметара за претраживање и преглед ТВ програма, тастер „Тражи“ и линкове за прелаз на остале странице. Претраживање реализовати по следећим пољима:

Датум – падајућа листа (DropDownList) са одговарајућом ознаком (Label) са предефинисаним вредностима,

Тип емисије - падајућа листа (DropDownList) са одговарајућом ознаком (Label) са предефинисаним вредностима

Поље за избор Датума реализовано падајућом листом напунити свим различитим датумима који су представљени фајловима у дефинисаном фолдеру у коме се чува ТВ програм.

Поље за избор Типа емисије реализовано падајућом листом напунити подацима о типовима емисија (нпр. Информативна емисија, Серија, Фilm...).

Резултат претраге представити у облику табеле која садржи три колоне:

- Време емитовања
- Емисија
- Тип емисије (графички приказ)

На страници „Контакт“ приказати текстуална поља за унос Имена, e-mail адресе, назива поруке и поруке преко које корисници могу евентуално да пошаљу коментаре и питања у виду порука, као и линкове за прелаз на остале странице. Функционалност слања порука није потребно реализовати.

На страници „Упутство“ написати кратко упутство за коришћење веб апликације и линкове за прелаз на остале странице.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације. Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Попуњене текстуалне датотеке које представљају програмске шеме.
- Фолдер са иконама типова емисија

За оцењивање користити образац за оцењивање број ЕИТ-А14

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А14 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ТВ ПРОГРАМ |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|----|----|----|----|----|----------------------|
| | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Поставке почетне странице

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Постављено поље са падајућом листом (DropDownList) „Датум“ | 3 | 0 |
| Постављено поље са падајућом листом (DropDownList) „Тип емисије“ | 3 | 0 |
| Постављен тастер „Тражи“ | 3 | 0 |
| Постављене одговарајуће ознаке (Label) поред поља за унос параметара претраге | 3 | 0 |
| Постављени ликови ка осталим страницама | 3 | 0 |

2. Поставке осталих страница

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Постављена страница за контакт | 4 | 0 |
| Постављена текстуална поља на страници за контакт | 5 | 0 |
| На страници за контакт постављени линкови ка осталим страницама | 3 | 0 |
| Постављена страница за преглед корисничког упутства | 5 | 0 |
| На страници за преглед корисничког упутства постављени линкови ка осталим страницама | 3 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 30) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Резултати претраге се приказују у табели испод параметара за претрагу | 7 | 0 |
| У поље са падајућом листом за датуме су учитани одговарајући подаци | 5 | 0 |
| У поље са падајућом листом за типове емисија су учитани одговарајући подаци | 5 | 0 |
| Функционише претрага по датумима и типовима емисија | 7 | 0 |
| Слика албума је приказана у резултатима претраге | 6 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 20) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Подаци су меморијски представљени у структуре или класе | 9 | 0 |
| CSS код се налази у посебном фајлу | 5 | 0 |
| Код за функционалност и изглед је у посебном фајлу | 6 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Веб апликација постављена на Веб сервер | 10 | 0 |
| Сви ресурси апликације су доступни | 5 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А15

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВОЗНИ ПАРК

ЕИТ-А15: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВОЗНИ ПАРК

Основни подаци о возилима неке компаније чувају се у текстуалној датотеци под називом „vozila.txt“, при чему се основни подаци о сваком возилу памте у једном реду текстуалне датотеке у следећем формату:

Регистарки број| Марка возила | Година производње | Боја

За свако возило треба да постоји и посебна датотека која за назив има регистарски број возила (нпр. NI123LE.txt) у којој се чувају подаци о пређеној километражи у следећем формату:

Датум | Од места | До места | Пређено километара

Креирати **MVC** веб апликацију користећи **Razor view engine** која води евиденцију о возилима компаније која треба да има следеће MVC View странице:

- Страницу за евидентирање пређеног пута сваког возила, при чему се подаци памте у предвиђеној датотеци која за назив има регистарски број возила.
- Страницу за приказ укупног пређеног пута за свако возило у виду табеле.

На страници за евидентирање пређеног пута возила поставити одговарајућа поља за избор возила (DropDownList које је напуњено подацима из текстуалне датотеке „vozila.txt“), унос датума (text input), унос стартне позиције (text input), унос крајње позиције (text input), колико је километара пређено (text input), тастер „Сними“ и линк ка страници за преглед укупног пређеног пута.

На страници за приказ укупног пређеног пута податке приказати у облику табеле која садржи следеће колоне:

- Регистарски број возила
- Марка возила
- Година производње
- Укупно пређено километара

Уз могућност филтрирања возила по регистарском броју и линк ка страници за евидентирање пређеног пута.

Акције снимања и читања података из текстуалних датотека реализовати у одговарајућим MVC контролерима. Податке меморијски представити одговарајућим MVC моделима.

CSS код издвојити у посебну екстерну датотеку, а учитати на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Попуњена текстуална датотека са подацима о возилима

За оцењивање користити образац за оцењивање број ЕИТ-А15

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

| | |
|-------------------------|--|
| Шифра радног задатка | ЕИТ-А15 |
| Назив радног задатка | КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВОЗНИ ПАРК |
| Назив школе | |
| Седиште | |
| Образовни профил | Електротехничар информационих технологија - оглед |
| Име и презиме кандидата | |
| Име и презиме ментора | |

| ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА: | | | | | | Укупно бодова |
|---|----|----|----|----|----|----------------------|
| | 1. | 2. | 3. | 4. | 5. | |
| Бодови | | | | | | |

| | |
|------------------------|----------------|
| Члан испитне комисије: | Место и датум: |
|------------------------|----------------|

| |
|-------------------|
| КОМЕНТАРИ: |
|-------------------|

1. Поставке почетне странице

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Постављено поље са падајућом листом (DropDownList) за избор возила | 4 | 0 |
| Постављена поља за унос података о пређеном путу | 3 | 0 |
| Постављен тастер „Сними“ | 3 | 0 |
| Постављене одговарајуће ознаке (Label) поред поља за унос података | 3 | 0 |
| Постављен линк за прелазак на другу страницу | 2 | 0 |

2. Поставке осталих страница

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Постављена табела за приказ укупног пређеног пута са одговарајућим колонама | 10 | 0 |
| Постављено поље за унос регистарског броја у циљу филтрирања података у табели | 3 | 0 |
| Линк за прелазак на страницу за евидентирање пређеног пута возила | 2 | 0 |

3. Функционалност веб апликације

| ИНДИКАТОРИ (максималан број бодова 30) | ТАЧНО | НЕТАЧНО |
|--|-------|---------|
| Сви захтевани подаци се снимају у текстуалну датотеку за евидентирање пређеног пута возила | 7 | 0 |
| У поље са падајућом листом за возила учитавају се одговарајући подаци | 5 | 0 |
| На страници за приказ укупног пређеног пута се приказују подаци о возилима и укупном пређеном путу | 5 | 0 |
| Функционише филтер по регистарском броју | 7 | 0 |
| Линкови између страница функционишу | 6 | 0 |

4. Организација података веб апликације

| ИНДИКАТОРИ (максималан број бодова 25) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Подаци су меморијски представљени у одговарајуће MVC моделе | 8 | 0 |
| Акције снимања и читања података из датотека налазе се у одговарајућим MVC контролерима | 7 | 0 |
| CSS код се налази у посебној датотеци | 5 | 0 |
| Код за изглед се налази у одговарајућим датотекама (MVC View) | 5 | 0 |

5. Поставке апликације на Веб серверу

| ИНДИКАТОРИ (максималан број бодова 15) | ТАЧНО | НЕТАЧНО |
|---|-------|---------|
| Веб апликација постављена на Веб сервер | 10 | 0 |
| Сви ресурси апликације су доступни | 5 | 0 |

ШИФРА ЗАДАТКА: ЕИТ-А16

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РАСПОРЕД ЧАСОВА

ЕИТ-А16: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РАСПОРЕД ЧАСОВА

Креирати **MVC** веб апликацију користећи **Razor view engine** за приказ и учитавање распореда часова неког одељења. Веб апликација треба да садржи следеће MVC View странице:

- Страницу за учитавање (upload) распореда часова из XML датотеке.
- Страницу за приказ учитаног распореда часова

На страници за учитавање распореда приказати одговарајуће поље (input) за избор XML датотеке са одговарајућом лабелом и тастер за иницирање акције учитавања датотеке у систем.

На страници за приказ распореда часова податке приказати у облику табеле која садржи следеће колоне:

- Р.бр. – редни број часа
- Понедељак
- Уторак
- Среда
- Четвртак
- Петак

У сваком реду табеле треба да се прикажу предмети који одговарају редном броју часа на задати дан у недељи. На ову страницу додати и линк ка страници за поновно учитавање распореда из XML датотеке.

Структура XML датотеке са рапоредом часова дата је у следећем формату:

```
<Raspored>
    <Rbr>1</Rbr>
    <DanUNedelji>Ponedeljak</ DanUNedelji>
    <Predmet>Matematika</Predmet>
</Raspored>
<Raspored>
    <Rbr>2</Rbr>
    <DanUNedelji>Ponedeljak</ DanUNedelji>
    <Predmet>Srpski</Predmet>
</Raspored>
<Raspored>
    <Rbr>3</Rbr>
    <DanUNedelji>Ponedeljak</ DanUNedelji>
    <Predmet>Programiranje</Predmet>
</Raspored>
```

Акције учитавања и читања података из XML датотеке реализовати у одговарајућим MVC контролерима. Податке меморијски представити одговарајућим MVC моделима.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталацији на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и приклучком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Попуњена XML датотека са распоредом часова

За оцењивање користити образац за оцењивање број ЕИТ-А16

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А16
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – РАСПОРЕД ЧАСОВА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Поставке почетнестранице

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Постављено поље за избор XML датотеке (input box)	9	0
Поставњен тастер „Учитај“	3	0
Постављена одговарајућа ознака(Label) поред поља за унос података	3	0

2. Поставке осталих страница

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Постављена табела за приказ распореда часова са одговарајућим колонама	12	0
Постављен линк за прелазак на страницу за поновно учитавање података из XML датотеке.	3	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 30)	ТАЧНО	НЕТАЧНО
Релизована функционалност избора и учитавања XML датотеке и подаци се преносе на серверску страну.	9	0
Врши се читање и десеријализација XML података из учитане датотеке	7	0
На страници за приказ распореда часова приказује се табела која садржи учитани рапоред часова	10	0
Линкови између страница функционишу	4	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 25)	ТАЧНО	НЕТАЧНО
Подаци су меморијски представљени одговарајућим MVC моделима	8	0
Акције учитавања и читања података из датотеке налазе се у одговарајућим MVC контролерима	7	0
CSS код се налази у посебној датотеци	5	0
Код за изглед се налази у одговарајућим датотекама (MVC View)	5	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	10	0
Сви ресурси апликације су доступни	5	0

ШИФРА ЗАДАТКА: ЕИТ-А17

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ ТЕСТ

ЕИТ-А17: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ ТЕСТ

Дата је XML датотека која садржи питања за електронски тест у следећем формату:

```
<Pitanje>
    <Rbr>1</Rbr>
    <Pitanje>Текст питања</ Pitanje>
    <Odgovor1>Понуђени одговор 1</Odgovor1>
    <Odgovor2>Понуђени одговор 2</Odgovor2>
    <Odgovor3>Понуђени одговор 3</Odgovor3>
    <Odgovor4>Понуђени одговор 4</Odgovor4>
    <Resenje>Тачан одговор</Resenje>
</Pitanje>
<Pitanje>
    <Rbr>2</Rbr>
    <Pitanje>Текст питања</ Pitanje>
    <Odgovor1>Понуђени одговор 1</Odgovor1>
    <Odgovor2>Понуђени одговор 2</Odgovor2>
    <Odgovor3>Понуђени одговор 3</Odgovor3>
    <Odgovor4>Понуђени одговор 4</Odgovor4>
    <Resenje>Тачан одговор</Resenje>
</Pitanje>
```

Креирати **MVC** веб апликацију користећи **Razor view engine** за приказ теста са 5 питања изабраних методом случајног избора и могућношћу слања одговора на сервер. Апликација треба да садржи следеће странице:

- Страницу за приказ питања са понуђеним одговорима
- Страницу за приказ одговора које је унео оператор

На страници за приказ питања исписати текст питања са понуђеним одговорима, приказати текстуално поље за унос одговора за свако питање и тастер за слање унетих одговора.

На страници за приказ одговора која се појављује након што корисник унесе одговоре и притисне тастер за слање одговора приказати тачне одговоре на питање, одговоре које је унео оператор и линк за поновни покушај тестирања.

Акције читања података из XML датотеке и слање одговора на сервер реализовати у одговарајућим MVC контролерима. Податке меморијски представити одговарајућим MVC моделима.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и приклjuчком на Интернет.

Време за израду задатка:

Максимално време предвиђено за израду овог задатка је 180 минута. Након овог времена извршење радног задатка се прекида, а не остварују се бодови за завршетак рада пре времена.

Прилози:

- Попуњена XML датотека са питањима и одговорима

За оцењивање користити образац за оцењивање број ЕИТ-А17

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А17
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ ТЕСТ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Поставке основне странице

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Постављена польза за приказ питања и понуђених одговора	6	0
Постављена текстуална польза за унос одговора	6	0
Постављен тастер за слање одговора	3	0
Постављена одговарајућа ознаке (Label) поред польза за унос и приказ података	2	0

2. Поставке осталих страница

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Постављена польза за приказ тачних одговора на питање	6	0
Постављена польза за приказ одговора које је унео оператор	6	0
Постављен линк за прелазак на страницу за поновни покушај тестирања	3	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 30)	ТАЧНО	НЕТАЧНО
Релизована функционалност читања (десеријализације) података из XML датотеке.	6	0
На страници се приказују питања са понуђеним одговорима	5	0
Одговори унети у текстуална польза се шаљу на веб сервер	6	0
На одговарајућој страници се приказују тачни одговори на питања	5	0
На одговарајућој страници се приказују одговори које је унео корисник	5	0
Линкови између страница функционишу	3	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 23)	ТАЧНО	НЕТАЧНО
Подаци су меморијски представљени одговарајућим MVC моделима	7	0
Акције слања података и читања података из датотеке имплементиране се у одговарајућим MVC контролерима	7	0
CSS код се налази у посебној датотеци	4	0
Код за изглед се налази у одговарајућим датотекама (MVC View)	5	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	10	0
Сви ресурси апликације су доступни	5	0

ШИФРА ЗАДАТКА: ЕИТ-А18

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА

ЕИТ-А18: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА

Дата је XML датотека која садржи податке о временској прогнози у следећем формату:

```
<Prognoza>
    <Mesto>001</Mesto>
    <NazivMesta>Ниш</NazivMesta>
    <MinTemperatura>0</MinTemperatura>
    <MaxTemperatura>10</MaxTemperatura>
    <Vreme>Ведро</Vreme>
</Prognoza>
<Prognoza>
    <Mesto>002</Mesto>
    <NazivMesta>Београд</NazivMesta>
    <MinTemperatura>-1</MinTemperatura>
    <MaxTemperatura>7</MaxTemperatura>
    <Vreme>Киша</Vreme>
</Prognoza>
<Prognoza>
    <Mesto>003</Mesto>
    <NazivMesta>Копаоник</NazivMesta>
    <MinTemperatura>-5</MinTemperatura>
    <MaxTemperatura>0</MaxTemperatura>
    <Vreme>Снег</Vreme>
</Prognoza>
```

Креирати **MVC** веб апликацију користећи **Razor view engine** за приказ података о временској прогнози прочитаних из наведене XML датотеке. Апликација треба да садржи следеће странице:

- Страницу за приказ временске прогнозе изабраног места
- Страницу за приказ максималних температура у свим местима

На страници за приказ временске прогнозе изабраног места поставити поље за избор места (DropDownList) које је напуњено свим местима која се налазе у XML датотеци, тастер „Прикажи“, поља за приказ података о временској прогнози, (Минимална температура и Максимална температура), сличицу која репрезентује податак о времену, (ведро, облачно, киша, снег) и линк на страницу за приказ максималних температура.

На страници за приказ максималних температура приказати у табеларном облику максималне температуре у свим местима прочитаних из XML датотеке, и линк на страницу за преглед временске прогнозе изабраног места.

Акције читања података из XML датотеке реализовати у одговарајућим MVC контролерима. Податке меморијски представити одговарајућим MVC моделима.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталан на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Попуњена XML датотека са подацима о временској прогнози
- Слике које репрезентују податак о времену (ведро, облачно, киша, снег)

За оцењивање користити образац за оцењивање број ЕИТ-А18

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А18
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ВРЕМЕНСКА ПРОГНОЗА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Поставке основне странице

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Постављена польза за избор места	5	0
Постављен тастер „Прикажи“	2	0
Постављена польза за приказ података о временској прогнози	3	0
Постављено пользе за приказ слике која репрезентују податак о времену	4	0
Постављене одговарајуће ознаке (Label) поред польза за унос и приказ података	2	0
Постављен линк за прелазак на страницу за приказ максималних температура	2	0

2. Поставке осталих страница

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Постављена табела за приказ максималних температура	10	0
Постављен линк за прелазак на страницу за приказ временске прогнозе	2	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Рализована функционалност читања (десеријализације) података из XML датотеке.	6	0
Пользе за избор места је попуњено подацима из XML датотеке	5	0
На страници се приказују подаци о временској прогнози	5	0
На страници се приказује слика која репрезентује податак о времену	6	0
На одговарајућој страници се приказује табела са максималним температурима у свим местима из XML датотеке	7	0
Линкови између страница функционишу	3	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 23)	ТАЧНО	НЕТАЧНО
Подаци су меморијски представљени одговарајућим MVC моделима	7	0
Акције претраге података и читања података из датотеке имплементиране се у одговарајућим MVC контролерима	7	0
CSS код се налази у посебној датотеци	4	0
Код за изглед се налази у одговарајућим датотекама (MVC View)	5	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	10	0
Сви ресурси апликације су доступни	5	0

ШИФРА ЗАДАТКА: ЕИТ-А19

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ РЕЧНИК

ЕИТ-А19: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ РЕЧНИК

Дата је база података која има једну табелу у којој се памте речи енглеског језика и њихов превод на српски језик. Табела има следећу структуру:

```
[ID] [bigint] IDENTITY(1,1) NOT NULL,  
[Engleski] [nvarchar](50) NOT NULL,  
[Srpski] [nvarchar](50) NOT NULL,  
[Opis] [nvarchar](1024) NULL
```

Креирати веб апликацију под називом “Електронски речник” која се састоји од две странице: Речник и странице за додавање нових речи у речник.

На страници „Речник“ поставити поља за избор смера превођења (Српско – Енглески, Енглеско - Српски), поље за унос речи за превођење, поља за приказ превода и описа, тастер „Преведи“ и линк за прелаз на страницу за додавање нових речи у речник. Поље за смер превођења реализовати у облику падајуће листе (DropDownList). Притиском на тастер „Преведи“ приказати одговарајући превод унете речи и опис уколико постоје у бази података.

На страници за додавање нових речи у речник поставити поља за унос енглеске речи, одговарајуће српске речи, поље за унос описа, тастер „Сними“ и линк за прелазак на страницу „Речник“. Притиском на тастер „Сними“ подаци се снимају у базу података.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- База података са предефинисаним скупом података

За оцењивање користити образац за оцењивање број ЕИТ-А19

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А19
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕЛЕКТРОНСКИ РЕЧНИК
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Поставке основне странице

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Постављена текстуална поља за унос речи за превод и приказ резултата превођења	6	0
Постављено поље са падајућом листом (DropDownList) за избор смера превођења	5	0
Постављен тастер „Преведи“	3	0
Постављене одговарајуће лабеле поред поља за унос и приказ података	3	0
Постављен линк ка страни за унос нових речи	3	0

2. Поставке осталих страница

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Постављена текстуална поља за унос нове речи, превода и описа	6	0
Постављен тастер „Сними“	3	0
Постављене одговарајуће лабеле поред поља за унос и приказ података	3	0
Постављен линк за прелазак на страницу „Речник“	3	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 30)	ТАЧНО	НЕТАЧНО
Реализована комуникација са базом података	4	0
Резултати превођења се приказују у предвиђеним пољима	10	0
У поље са падајућом листом су учитани одговарајући смерови превођења	5	0
Нове речи, превод и опис се памте у бази података	8	0
Линкови између страница функционишу	3	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Подаци су меморијски представљени у структуре или класе	9	0
CSS код се налази у посебном фајлу	5	0
Обрађени изузети приликом комуникације са базом података	6	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	10	0
Сви ресурси апликације су доступни	5	0

ШИФРА ЗАДАТКА: ЕИТ-А20

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СЛАЈДЕР ЗА СЛИКЕ (Image Slider)

ЕИТ-А20: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СЛАЈДЕР ЗА СЛИКЕ (IMAGE SLIDER)

Дат је фолдер са slikama и следећа HTML структура на веб страници:

```
<div id="slike">
    </img>
    </img>
    </img>
    </img>
    </img>
</div>
```

Креирати веб апликацију са обрадом на клијентској страни под називом “Image slider”, која од постојеће структуре креира панел на коме се смењују слике једна за другом у временском интервалу од 3 секунде (3000 ms). Након приказа последње слике у низу приказује се поново прва слика. Испод слике која се тренутно приказује креирати индикатор у виду квадратића који приказује која слика у низу је тренутно приказана. Приликом приказа слике коришћењем CSS особине „*opacity*“, креирати ефекат „*FadeIn*“, где слика од потпуно провидне, постепено, (у року од 1000 ms), постане потпуно видљива. На овој страници потребно је креирати и линк до странице „О апликацији“ на којој се налазе информације о аутору и апликацији (Име, презиме и email ученика, име апликације и датум креирања).

На страници „О апликацији“ се налазе горе наведени подаци и линк за повратак на страницу где се налази слайдер са slikama.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације. Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

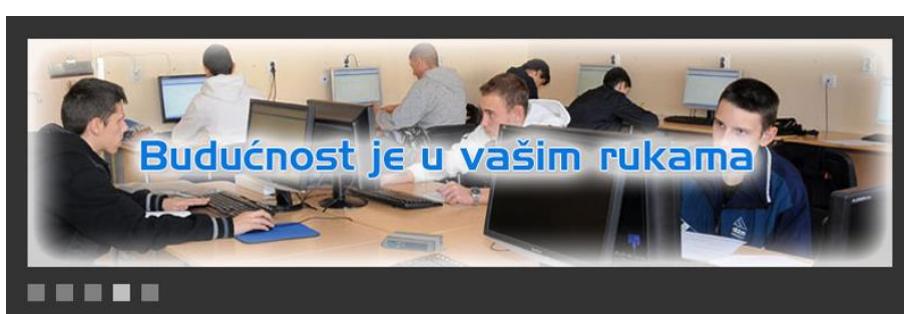
Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- Фолдер са slikama које се приказују на слайдеру за слике
- Пример слайдера



За оцењивање користити образац за оцењивање број ЕИТ –А20

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А20
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СЛАЈДЕР ЗА СЛИКЕ (Image slider)
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Поставке основне странице

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Постављен панел за приказ слика	10	0
Постављени индикатори који показују која слика у низу је тренутно приказана	7	0
Постављен линк ка страни „О апликацији“	3	0

2. Поставке осталих страница

ИНДИКАТОРИ (максималан број бодова 10)	ТАЧНО	НЕТАЧНО
Постављена текстуална поља за приказ података о аутору и апликацији	7	0
Постављен линк за провратак на основну страницу	3	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 35)	ТАЧНО	НЕТАЧНО
На панелу се приказују слике дефинисане у задатку	10	0
Слике се смењују на задати временски интервал	9	0
После приказа последње слике приказује се поново прва слика	6	0
Креиран ефекат „FadeIn“ приликом приказа слика	7	0
Линкови између страница функционишу	3	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Појединачне операције (иницијализација, промена слика, Fadeln ефекат) су организоване као посебне функције	9	0
CSS код се налази у посебном фајлу	5	0
Код за функционалност је у посебном фајлу	6	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	10	0
Сви ресурси апликације су доступни	5	0

ШИФРА ЗАДАТКА: ЕИТ-А21

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СОРТИРАЊЕ ТАБЕЛЕ

ЕИТ-А21: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – СОРТИРАЊЕ ТАБЕЛЕ

Дата је HTMLтабела са три колоне (Рбр, Име, Презиме) на веб странициса следећим подацима:

```
<table id="MojaTabela">
    <tr>
        <th><a href="#">Rbr</a></th>
        <th><a href="#">Ime</a></th>
        <th><a href="#">Prezime</a></th>
    </tr>
    <tr>
        <td>1.</td>
        <td>Pera</td>
        <td>Mikic</td>
    </tr>
    <tr>
        <td>2.</td>
        <td>Mika</td>
        <td>Zikic</td>
    </tr>
    <tr>
        <td>3.</td>
        <td>Zika</td>
        <td>Peric</td>
    </tr>
</table>
```

Креирати веб апликацију са обрадом на клијентској страни под називом “Сортирање табеле” која има функционалност сортирања табеле по свим колонама у опадајући или растући редослед. Акција сортирања табеле у растући редослед по жељеној колони се иницира левим кликом миша на заглавље (header) те колоне. Сортирање у опадајући редослед се иницира поновним кликом миша на заглавље колоне. Сваким наредним кликом миша на заглавље колоне наизменично се мења сортирање у опадајући или растући редослед те колоне. Заглавље табеле нагласити другом бојом коришћењем CSS описа. На овој страници потребно је креирати и линк до странице „О апликацији“, на којој се налазе информације о аутору и апликацији (Име, презиме и email ученика, име апликације и датум креирања).

На страници „О апликацији“ се налазе горе наведени подаци и линк за повратак на страницу, где се налази табела која има могућност сортирања.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- HTML табела са подацима
- Пример табеле:

Rbr	Ime	Prezime
1.	Pera	Mikic
2.	Mika	Zikic
3.	Zika	Peric

За оцењивање користити образац за оцењивање број ЕИТ –А21

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А21
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ –СОРТИРАЊЕ ТАБЕЛЕ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Поставке основне странице

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Постављена табела са подацима	10	0
Наглашено заглавље табеле коришћењем CSS описа	7	0
Постављен линк ка страни „О апликацији“	3	0

2. Поставке осталих странице

ИНДИКАТОРИ (максималан број бодова 10)	ТАЧНО	НЕТАЧНО
Постављена текстуална поља за приказ података о аутору и апликацији	7	0
Постављен линк за првратак на основну страницу	3	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 35)	ТАЧНО	НЕТАЧНО
Табела се сортира у растући редослед по 1. колони левим кликом миша на заглавље колоне	5	0
Табела се сортира у опадајући редослед по 1. колони левим кликом миша на заглавље колоне	5	0
Табела се сортира у растући редослед по 2. колони левим кликом миша на заглавље колоне	5	0
Табела се сортира у опадајући редослед по 2. колони левим кликом миша на заглавље колоне	5	0
Табела се сортира у растући редослед по 3. колони левим кликом миша на заглавље колоне	5	0
Табела се сортира у опадајући редослед по 3. колони левим кликом миша на заглавље колоне	5	0
Сваким наредним кликом миша на заглавље колоне наизменично се смењује сортирање у опадајући и растући редослед	2	0
Линкови између страница функционишу	3	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Појединачне операције (иницијализација и сортирање података) су организоване као посебне функције	9	0
CSS код се налази у посебном фајлу	5	0
Код за функционалност је у посебном фајлу	6	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	10	0
Сви ресурси апликације су доступни	5	0

ШИФРА ЗАДАТКА: ЕИТ-А22

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕФЕКТИ ТАБЕЛЕ

ЕИТ-А22: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕФЕКТИ ТАБЕЛЕ

Дата је HTML табела са списком ученика која има пет колона (Рбр, Име, Презиме, Школа, Поени) као на следећем примеру:

```
<table id="Tabela">
    <tr>
        <th>Rbr</th>
        <th>Ime</th>
        <th>Prezime</th>
        <th>Škola</th>
        <th>Poeni</th>
    </tr>
    <tr>
        <td>1.</td>
        <td>Pera</td>
        <td>Mikic</td>
        <td>OŠ "Vuk Karadžić"</td>
        <td>54</td>
    </tr>
    <tr>
        <td>2.</td>
        <td>Mika</td>
        <td>Zikic</td>
        <td>OŠ "Ivo Andrić"</td>
        <td>42</td>
    </tr>
    ...
</table>
```

Креирати веб апликацију са обрадом на клијентској страни под називом „Ефекти табеле“ која на основној страници има постављену наведену табелу и испод табеле постављене следеће тастере:

- Сортирај – сортира сетабела у опадајућем редоследу по колони „Поени“
- Обоји алтернативне редове – Позадина непарних редова боји се у сиво
- Маркирај оне који су положили – Боји се текст у редовима где ученици имају више од 50 поена у зелено
- Маркирај оне који нису положили – Боји се текст у редовима где ученици имају мање од 50 поена у црвено
- Издвоји оне који су положили – Из табеле се избацију сви ученици који имају мање од 50 поена, а затим се преостали редови сортирају у опадајућем редоследу по колони „Поени“.

Заглавље табеле нагласити другом бојом коришћењем CSS описа. На овој страници потребно је креирати и линк до странице „О апликацији“ на којој се налазе информације о аутору и апликацији (Име, презиме и email ученика, име апликације и датум креирања).

На страници „О апликацији“ се налазе горе наведени подаци и линк за повратак на страницу где се налази табела која има могућност сортирања.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- HTML табела са подацима

За оцењивање користити образац за оцењивање број ЕИТ –А22

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А22
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЕФЕКТИ ТАБЕЛЕ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Поставке странице са табелом

ИНДИКАТОРИ (максималан број бодова 25)	ТАЧНО	НЕТАЧНО
Постављена табела са подацима	10	0
Постављени тастери за извршење акција над табелом	5	0
Наглашено заглавље табеле коришћењем CSS описа	7	0
Постављен линк ка страни „О апликацији“	3	0

2. Поставке странице „О апликацији“

ИНДИКАТОРИ (максималан број бодова 10)	ТАЧНО	НЕТАЧНО
Постављена текстуална поља за приказ података о аутору и апликацији	7	0
Постављен линк за превратак на основну страницу	3	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 30)	ТАЧНО	НЕТАЧНО
Притиском на тастер „Сортирај“ табела се сортира у опадајућем редоследу по колони „Поени“	5	0
Притиском на тастер „Обоји алтернативне редове“ позадина непарних редова боји се у сиво	5	0
Притиском на тастер „Маркирај оне који су положили“ боји се текст редова код ученика који имају више од 50 поена у зелено	5	0
Притиском на тастер „Маркирај оне који нису положили“ боји се текст редова код ученика који имају мање од 50 поена у црвено	5	0
Притиском на тастер „Издвоји оне који су положили“ из табеле се избацују сви ученици који имају мање од 50 поена, а затим се преостали редови сортирају у опадајућем редоследу по колони „Поени“	7	0
Линкови између страница функционишу	3	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Појединачне операције (иницијализација, сортирање података, бојења редова) су организоване као посебне функције	9	0
CSS код се налази у посебном фајлу	5	0
Код за функционалност је у посебном фајлу	6	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	10	0
Сви ресурси апликације су доступни	5	0

ШИФРА ЗАДАТКА: ЕИТ-А23

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КАРТИЦЕ ЛОЈАЛНОСТИ

ЕИТ-А23: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КАРТИЦЕ ЛОЈАЛНОСТИ

Дата је база података која има једну табелу у којој се памте подаци о картици лојалности (Loyalty cards). Табела има следећу структуру:

```
[BrojKartice] [nvarchar](10) NOT NULL,  
[ImeVlasnika] [nvarchar](50) NOT NULL,  
[PrezimeVlasnika] [nvarchar](50) NOT NULL,  
[AdresaVlasnika] [nvarchar](50) NULL  
[OstvareniBodovi] [float] NOTNULL  
[OstvareniPopust] [float] NOTNULL  
[RokVazenja] [float] NOTNULL
```

Креирати веб апликацију која има страницу на којој се налази поље за унос броја картице, тастер „Провера картице“ и поља за приказ података о картици који суочитани из базе уколико картица са унетим бројем постоји у бази. У случају да картица не постоји у бази података, или је неважећа, обавестити оператора у виду поруке обојене црвеном бојом. Комуникацију између клијента и сервера реализацијем jQuery AJAX методе тако да при комуникацији са сервером не постоји потреба за поновним учитавањем целе веб странице. На овој страници потребно је креирати и линк до странице „О апликацији“, на којој се налазе информације о аутору и апликацији (Име, презиме и email ученика, име апликације и датум креирања).

Комуникацију са базом података и обраду реализовати на серверској страни.

На страници „О апликацији“ се налазе горе наведени подаци и линк за повратак на страницу где се налази табела која има могућност сортирања.

CSS код издвојити у посебну екстерну датотеку и учитати је на основну страницу при стартовању апликације.

Готову апликацију поставити на веб сервер инсталан на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилози:

- База података са предефинисаним скупом података

За оцењивање користити образац за оцењивање број ЕИТ –А23

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А23
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – КАРТИЦЕ ЛОЈАЛНОСТИ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Поставке основне странице

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Постављено текстуално поље за унос броја картице	5	0
Постављен тастер за иницирање акције провере картице	5	0
Постављена поља за приказ података о картици	7	0
Постављен линк ка страни „О апликацији“	3	0

2. Поставке осталих страница

ИНДИКАТОРИ (максималан број бодова 10)	ТАЧНО	НЕТАЧНО
Постављена текстуална поља за приказ података о аутору и апликацији	7	0
Постављен линк за првратак на основну страницу	3	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 35)	ТАЧНО	НЕТАЧНО
Притиском на тастер „Провера картице“ иницира се JQuery AJAX метода за прибављање података од сервера	10	0
Реализована комуникација са базом и читање података из базе према задатом упиту	10	0
Подаци се шаљу клијенту и приказују у одговарајућа поља	7	0
Приказује се упозорење уколико картица не постоји или је неважећа	5	0
Линкови између страница функционишу	3	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Подаци на серверској страни су меморијски представљени у структуре или класе	9	0
CSS код се налази у посебном фајлу	5	0
Обрађени изузети приликом комуникације са базом података	6	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	10	0
Сви ресурси апликације су доступни	5	0

ШИФРА ЗАДАТКА: ЕИТ-А24

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – УСПЕХ УЧЕНИКА

ЕИТ-А24: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – УСПЕХ УЧЕНИКА

Креирати веб апликацију са називом “Успех ученика” која се састоји од три странице: Почетна, Графика и О аутору .

Подаци о успеху ученика по разредима налазе се у бази података **Skola**, која има само једну табелу **Uspeh**, са следећим атрибутима:

- Razred (број разреда – целобројни без децимала)
- Odlican (број одличних ученика – целобројни без децимала)
- Vrlodobar (број врло добрих ученика – целобројни без децимала)
- Dobar (брво добрих ученика – целобројни без децимала)
 - Dovoljan (брво довољних ученика – целобројни без децимала)
 - Nedovoljan (брво недовољних ученика – целобројни без децимала)
 - ProsOcena (просечна оцена успеха разреда – децимални број са две децимале)

На страници **Почетна**, табеларно приказати податке о успеху ученика сортиране по просечној оцени коришћењем GridView контроле.

На страници **Графика** графички приказати податке о успеху ученика у облику Chart Type- Column. На x-оси представити податке из поља Razred, а на y-оси податке из поља ProsOcena.

На страници **О аутору** написати основне податке о аутору.

CSS код издвојити у посебну екстерну датотеку, а учитати је при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталан на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилог:

- Табела **Uspeh** попуњена подацима о успеху ученика.

За оцењивање користити образац за оцењивање број ЕИТ –А24

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А24
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ –УСПЕХ УЧЕНИКА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Креирање веб страница

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Креиране све веб странице према задатку	5	0
Странице имају одговарајући садржај	6	0
Функционишу све везе између страница	8	0

2. Поставке почетне веб странице

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Постављена тражена форма на страницу	5	0
Форма има све елементе према задатку	8	0
Подаци на форми су приказани према задатку	8	0

3. Поставке странице Графика

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Постављена тражена форма на страницу	5	0
Форма има све елементе према задатку	8	0
Подаци на форми су приказани према задатку	8	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Реализована веза са базом података	5	0
CSS код и код за функционалност су у посебним фајловима	7	0
Кодови за функционалност и CSS се учитавају при стартовању	8	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	5	0
Апликација се покреће позивом фајла index.html	6	0
Сви линкови у апликацији правилно функционишу	8	0

ШИФРА ЗАДАТКА:**ЕИТ-А25**

НАЗИВ ЗАДАТКА: **КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – БИБЛИОТЕКА**

ЕИТ-А25: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – БИБЛИОТЕКА

Креирати веб апликацију са називом “Библиотека” која се састоји од три странице: **Почетна**, **Прегледи О аутору**.

Подаци о функционисању библиотеке налазе се XML фајлу biblioteka.xml у следећем облику:

```
<biblioteka>
  <knjiga ISBN="10-000000-001"
    naslov="Baze podataka"
    stanje="12"
    citano="21">
  </knjiga>

  <knjiga ISBN="10-000000-002"
    naslov="Programiranje C#"
    stanje="24"
    citano="21">
  </knjiga>

  <knjiga ISBN="11-000000-003"
    naslov="Web programiranje"
    stanje="5"
    citano="21">
  </knjiga>
</biblioteka>
```

На страници **Почетна** поставити форму за логовање која треба да омогући логовање на сајт за најмање два корисничка имена и шифре. По логовању исписати текст: „Добродошли на страницу!“

На страници **Преглед** табеларно приказати податке из фајла biblioteka.xml, сортиране према најчитанијој књизи.

На страници **О аутору** написати основне податке о аутору.

CSS код издвојити у посебну екстерну датотеку, а учитати је при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталан на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилог:

- Датотека biblioteka.xml попуњена подацима

За оцењивање користити образац за оцењивање број ЕИТ-А25

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А25
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ –БИБЛИОТЕКА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Креирање веб страница

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Креиране све веб странице према задатку	5	0
Странице имају одговарајући садржај	6	0
Функционишу све везе између страница	8	0

2. Поставке почетне веб странице

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Постављена тражена форма на страницу	5	0
Форма има све елементе према задатку	8	0
Логовање на сајт функционише према задатку	8	0

3. Поставке странице Преглед

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Постављена тражена форма на страницу	5	0
Форма има све елементе према задатку	8	0
Подаци на форми су сортирани према задатку	8	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Реализована веза са базом података	5	0
CSS код и код за функционалност су у посебним фајловима	7	0
Кодови за функционалност и CSS се учитавају при стартовању	8	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	5	0
Апликација се покреће позивом фајла index.html	6	0
Сви линкови у апликацији правилно функционишу	8	0

ШИФРА ЗАДАТКА: ЕИТ-А26

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЗВУЦИ ИНСТРУМЕНТА

ЕИТ-А26: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЗВУЦИ ИНСТРУМЕНТА

Креирати веб апликацију са називом “Звуци инструмената” која се састоји од три странице: **Почетна, О аутору и Упутство**. Апликација је намењена за ученике основне школе и потребе предмета Музичка култура.

На страници Почетна поставити табелу која треба да има три колоне у следећем облику.

Назив инструмента	Слика инструмента	Звук инструмента

У колонама поставити респективно: назив инструмента, слику инструмента у jpg формату и слику звучника у jpg формату. Потребно је попунити најмање 6 врста података у табели.

На страници О аутору написати основне податке о аутору.

На страници Упутство написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: Левим кликом миша на слику инструмента отвара се нова страница која нас води на сайт <http://sr.wikipedia.org/> где се налази опис тог инструмента. Преласком миша преко слике звучника чује се карактеристичан звук тог инструмента.

CSS код и код за функционалност издвојити у посебне екстерне датотеке, а учитати их у HTML при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталiran на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прилог:

- Слике и звуци за најмање шест инструмената.

За оцењивање користити образац за оцењивање број ЕИТ –А26

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А26
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – ЗВУЦИ ИНСТРУМЕНТА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Креирање веб страница

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Креиране све веб странице према задатку	5	0
Странице имају одговарајући садржај	6	0
Функционишу све везе између страница	8	0

2. Поставке почетне веб странице

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Постављена табела на веб страницу	5	0
Постављене слике на веб страницу	6	0
Позадина и слике одговарају захтевима задатка	8	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Левим кликом миша на слику отвара се нова страница	6	0
Страница има одговарајући садржај	6	0
Преласком миша преко слике звучника чује се тражени звук	7	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Слике и фајлови се налазе у посебним фолдерима	5	0
CSS код се налази у посебном фајлу	5	0
Код за функционалност је у посебном фајлу	6	0
Кодови за функционалност и CSS се учитавају при стартовању	8	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	5	0
Апликација се покреће позивом фајла index.html	6	0
Сви линкови у апликацији правилно функционишу	8	0

ШИФРА ЗАДАТКА: ЕИТ-А27

НАЗИВ ЗАДАТКА: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – БИОРИТАМ

ЕИТ-А27: КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ – БИОРИТАМ

Креирати веб апликацију са називом “Биоритам” која се састоји од три странице: **Почетна**, **О аутору** и **Упутство**. Апликација је намењена за забаву и заснива се на теорији биоритма која није научно доказана.

На страници **Почетна** поставити форму за унос два датума: Датума рођења и Датум биоритма. Испод форме поставити дугме са текстом „*Prikazi bioritam*“.

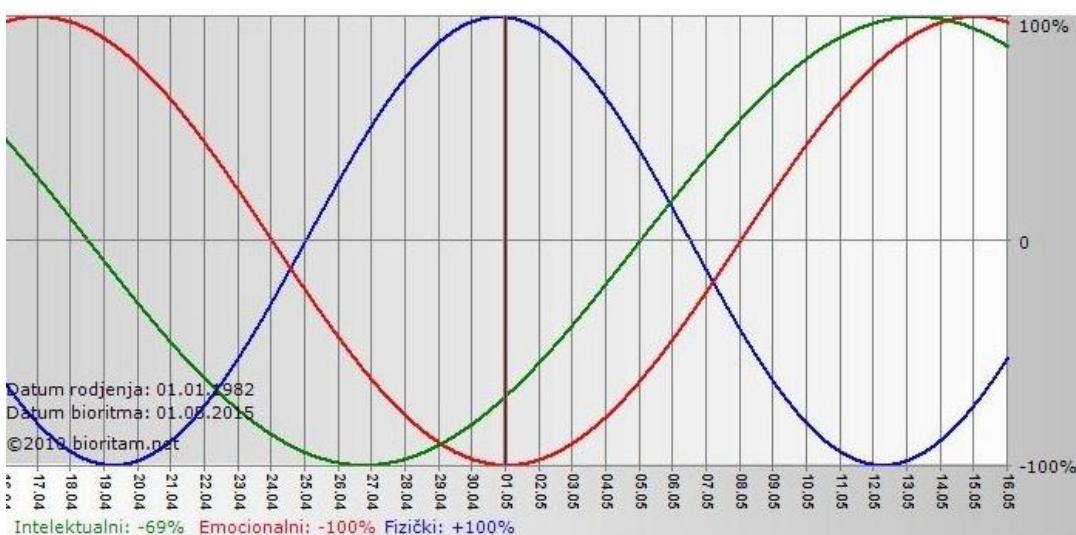
Теорија биоритма заснива се на уверењу да се људски организам и психа мењају у циклусима, а заједнички тренутак им је рођење. У тренутку кад се роди, човек пролази кроз различите емотивне, физичке и интелектуалне фазе. Постоје два различита циклуса и три биоритма који се мере.

Биоритми су физички, емоционални и интелектуални. Након рођења све су криве на нултој тачки, а успони и падови се понављају. Физички свака 23 дана, емоционални сваких 28 дана, а интелектуални свака 33 дана. Биоритам се приказује синусним функцијама које имају нуле у тренутку рођења. Циклуси биоритма су подељени у два дела, при чему се фазе посматрају као активне и пасивне или негативне и позитивне.

На страници **О аутору** написати основне податке о аутору.

На страници **Упутство** написати кратко упутство за коришћење веб апликације.

Веб апликација треба да има следећу функционалност: Унети тражене датуме. Притиском на дугме „*Prikazi bioritam*“ израчунати број дана од рођења до траженог датума као и податке за сваки циклус. Графички приказати податке за сваки циклус у различитим бојама као што је приказано на слици за 15 дана пре и 15 дана после жељеног датума.



CSS код и код за функционалност издвојити у посебне екстерне датотеке, а учитати их у HTML при стартовању апликације.

Готову апликацију поставити на Веб сервер инсталацијан на локалном рачунару.

Прибор за рад:

Рачунар са потребним програмима и прикључком на Интернет.

Време за израду задатка:

Максимално време за израду задатка је 180 минута. По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

За оцењивање користити образац за оцењивање број ЕИТ-А27

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-А27
Назив радног задатка	КРЕИРАЊЕ ВЕБ АПЛИКАЦИЈЕ –БИОРИТАМ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Креирање веб страница

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Креиране све веб странице према задатку	5	0
Странице имају одговарајући садржај	6	0
Функционишу све везе између страница	8	0

2. Поставке почетне веб странице

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Постављена форма за унос датума	5	0
Постављено дугме на веб страницу	6	0
Постављена контрола за графички приказ биоритма	8	0

3. Функционалност веб апликације

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Подаци са форме се правилно прихватају	5	0
Број дана између датума као и циклуси се правилно израчунавају	5	0
Подаци на контроли за графички приказ се правилно исцртавају	7	0
Апликација потпуно функционише према условима задатка	7	0

4. Организација података веб апликације

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
CSS код се налази у посебном фајлу	5	0
Код за функционалност је у посебном фајлу	6	0
Кодови за функционалност и CSS се учитавају при стартовању	8	0

5. Поставке апликације на Веб серверу

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Веб апликација постављена на Веб сервер	5	0
Апликација се покреће позивом фајла index.html	6	0
Сви линкови у апликацији правилно функционишу	8	0

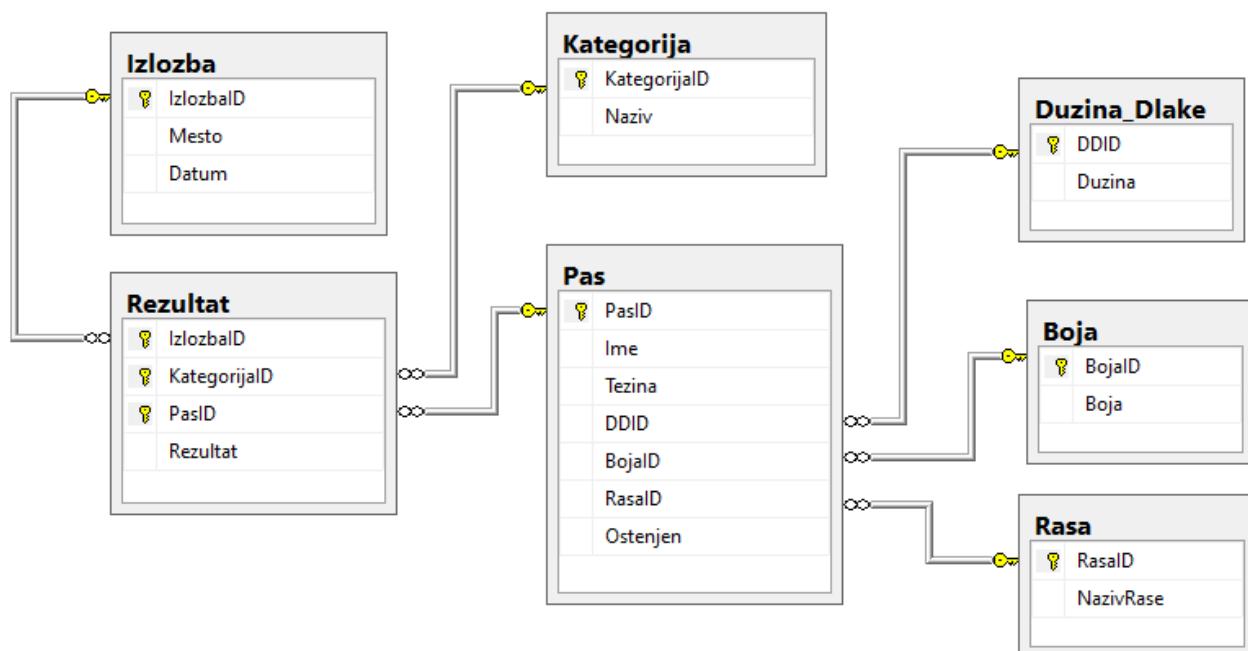
КОМПЕТЕНЦИЈА Б

ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б1
НАЗИВ РАДНОГ ЗАДАТКА: ИЗЛОЖБЕ ПАСА

ЕИТ-Б1: ИЗЛОЖБЕ ПАСА

Креирана је база података у којој се води евиденцију о изложбама паса које се одржавају у Србији.

Дијаграм базе је дат на слици:



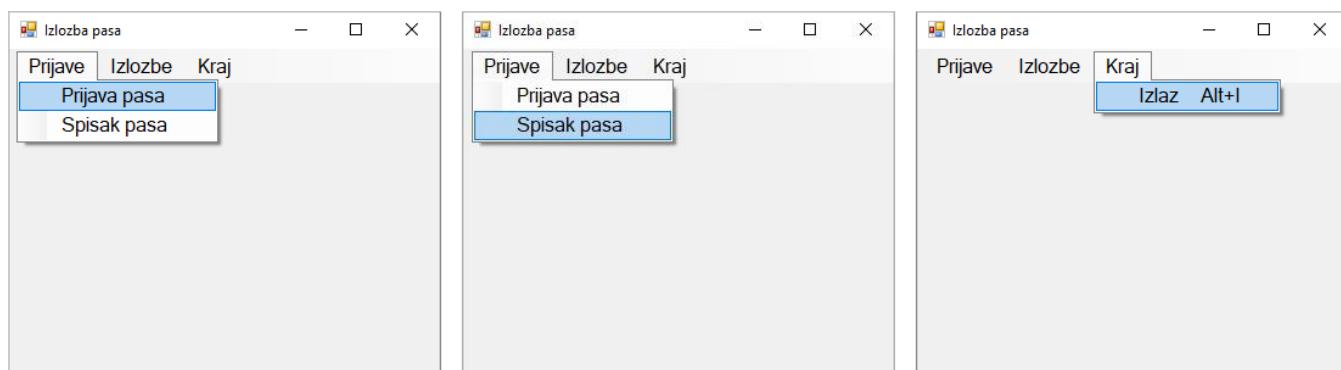
Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

У табели резултат треба додати и напомену у којој ће се чувати додатна запажања о псу који је остварио одређени резултат.

Пас може имати само једног власника, а власник може имати више паса. За власника је потребно чувати поред података о имену и презимену, адресу и матични број.

За сваког пса потребно је чувати и пол у посебној табели.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о пријавама обавља се преко обрасца чији је изглед дат на слици:

Izbor akcije
 Unos Brisanje

Pas: 17 - Bianca

Izložba: NIS072014 - Niš

Kategorija: 2 - CACIB

Upši Obriši Izađi

Унос пријаве

Izbor akcije
 Unos Brisanje

Pas: 19 - Leksi

Izložba: KRA052015 - Kraljevo

Kategorija: 1 - Mladi

Upši Obriši Izađi

Брисање пријаве

За пса се приказују шифра и назив сортирани у растући низ.

За изложбу се приказују шифра и место сортирани у растући низ.

За категорију се приказују шифра и назив сортирани у растући низ.

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

При брисању изабраног пса, бришу се све његове пријаве (за све изложбе и категорије за које је био пријављен).

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Spisak pasa**, приказује колико се такмичило паса у свакој категорији за одређену изложбу.

Образац треба да изгледа као на слици:

Izložba: NOV052015 - Novi Sad

Ukupan broj pasa koji je prijavljen: 64

Ukupan broj pasa koji se takmičio: 60

	Šifra	Naziv kategorije	Broj pasa
▶	1	Mladi	12
	2	CACIB	13
	3	Srednji	13
<	4	Starii	12
			▼

Prikaži Izađi

Подаци се приказују у контроли за табеларни приказ података.

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б1

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б1				
Назив радног задатка	ИЗЛОЖБЕ ПАСА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 22)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за пса	5	0
Постављена иницијална вредност за изложбу	5	0
Постављена иницијална вредност за категорију	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 26)	ТАЧНО	НЕТАЧНО
Обезбеђена контрола избора акције за брисање/упис	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	10	0
Обезбеђено брисање података из базе по спецификацији задатка	8	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за шифру изложбе	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 25)	ТАЧНО	НЕТАЧНО
Приказани подаци у контроли за табеларни приказ података	11	0
Приказан укупан број паса који су пријављени	5	0
Приказан укупан број паса који се такмично	5	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

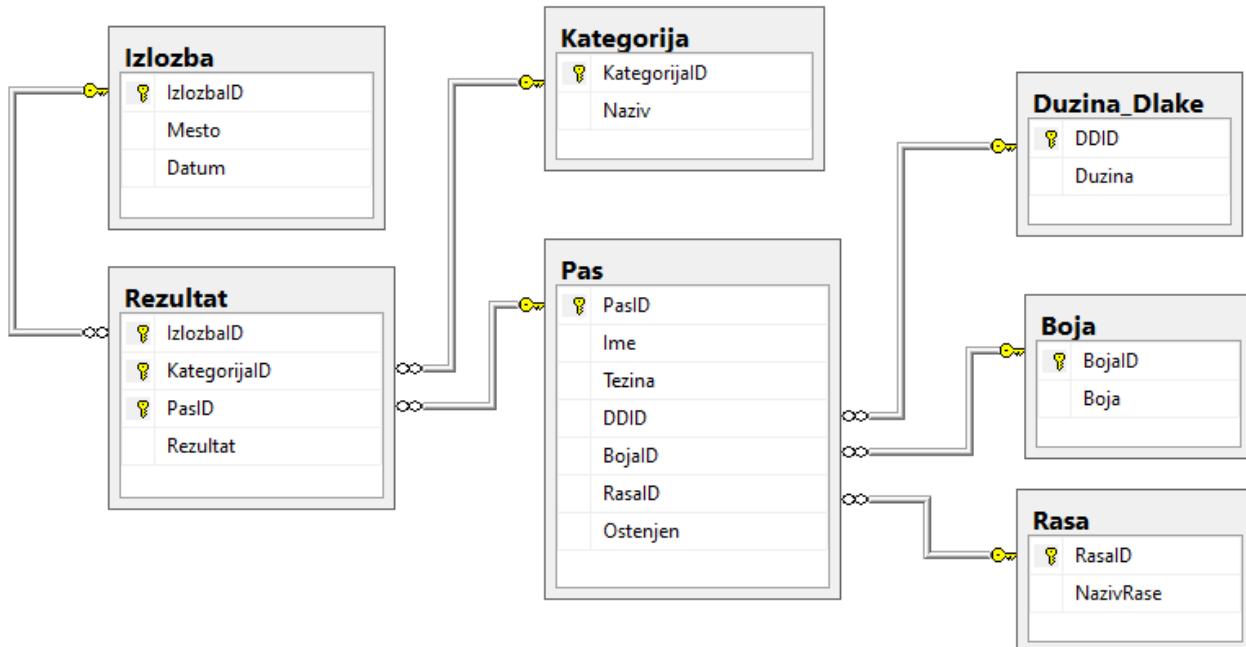
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б2

НАЗИВ РАДНОГ ЗАДАТКА: ИЗЛОЖБЕ ПАСА

ЕИТ-Б2: ИЗЛОЖБЕ ПАСА

Креирана је база података у којој се води евиденцију о изложбама паса које се одржавају у Србији.

Дијаграм базе је дат на слици:



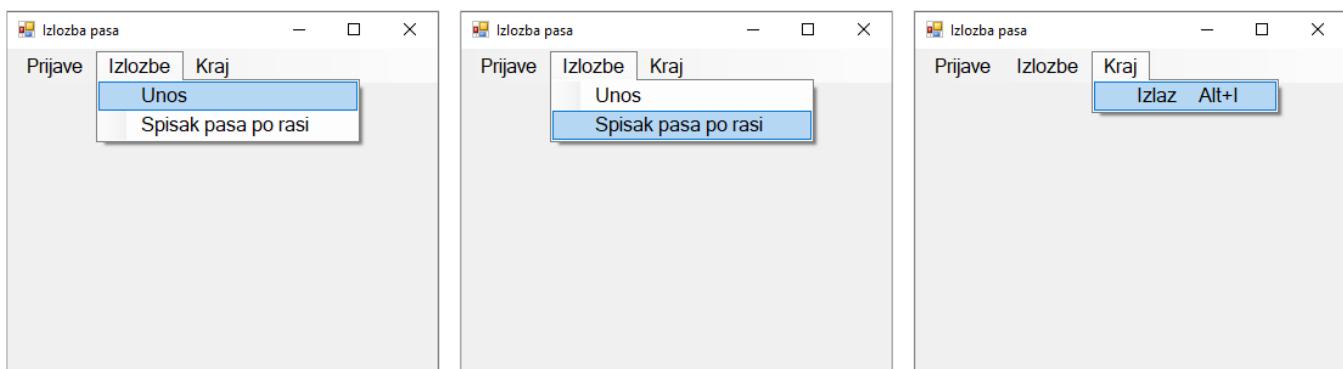
Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

У табели резултат треба додати и напомену у којој ће се чувати додатна запажања о псу који је остварио одређени резултат.

Пас може имати само једног власника, а власник може имати више паса. За власника је потребно чувати поред података о имену и презимену, адресу и матични број.

За сваког пса потребно је чувати и пол у посебној табели.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о изложбама обавља се преко обрасца чији је изглед дат на слици:

Упис изложбе

Брисање изложбе

Шифра изложбе се аутоматски формира као реч од 9 слова на следећи начин: прва три слова места, месец и година одржавања изложбе (нпр. **Beograd, 14.05.2015** даје шифру **BEO052015**).

Дугме **Izadjii** затвара отворени образац, и враћа се на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Spisak pasa po rasi**, приказује листу имена свих паса тражене расе који су учествовали на траженој изложби сортирани по имениу у опадајући низ.

Образац треба да изгледа као на слици:

За изложбу се приказује шифра сортирана у растући низ.

За расе се приказује назив сортиран у растући низ.

Подаци се приказују у контроли за табеларни приказ података.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б2

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б2
Назив радног задатка	ИЗЛОЖБЕ ПАСА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 7)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Обезбеђена контрола избора акције за брисање/упис	4	0
Формирана шифра изложбе по спецификацији задатка	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	11	0
Обезбеђено брисање података из базе по спецификацији задатка	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 14)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за шифру изложбе	5	0
Постављена иницијална вредност за расу пса	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 29)	ТАЧНО	НЕТАЧНО
Приказани подаци у контроли за табеларни приказ података	15	0
Приказан укупан број паса који су пријављени	5	0
Приказан укупан број паса који се такмично	5	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

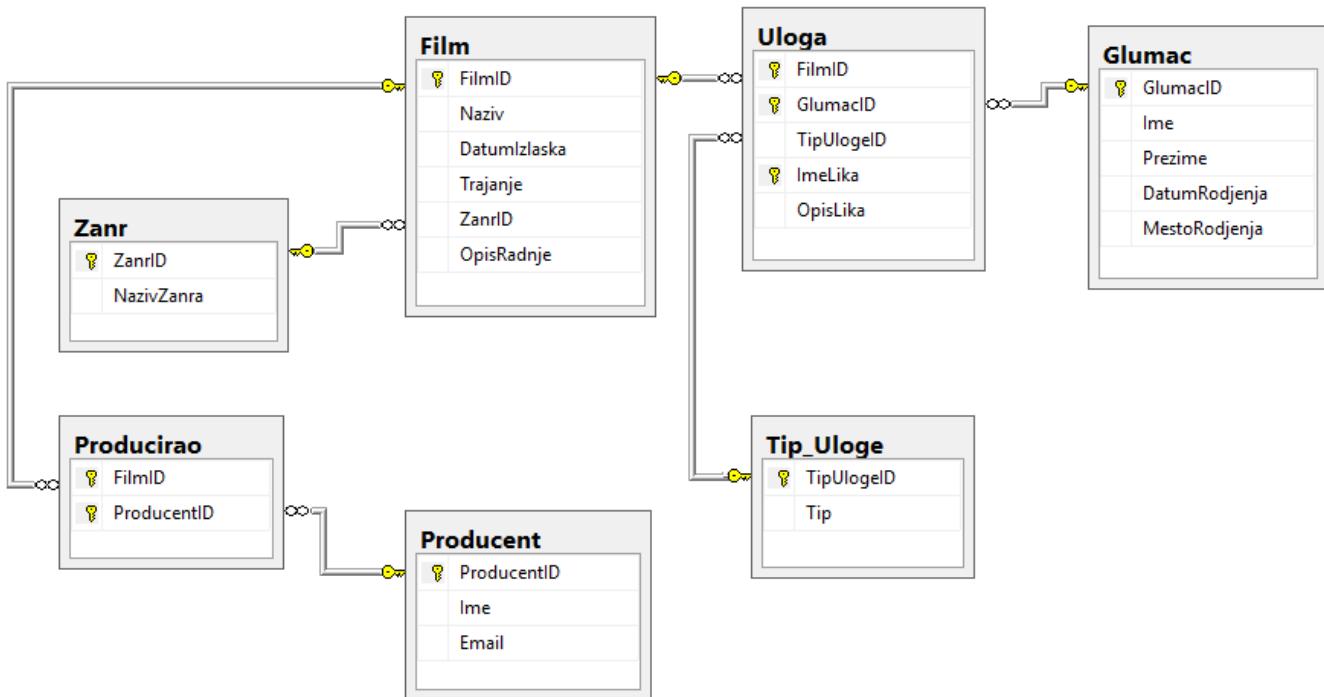
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б3

НАЗИВ РАДНОГ ЗАДАТКА: ДВД КОЛЕКЦИЈА

ЕИТ-Б3: ДВД КОЛЕКЦИЈА

Креирана је база података у којој се чувају подаци о твојој личној ДВД колекцији:

Дијаграм базе је дат на слици:

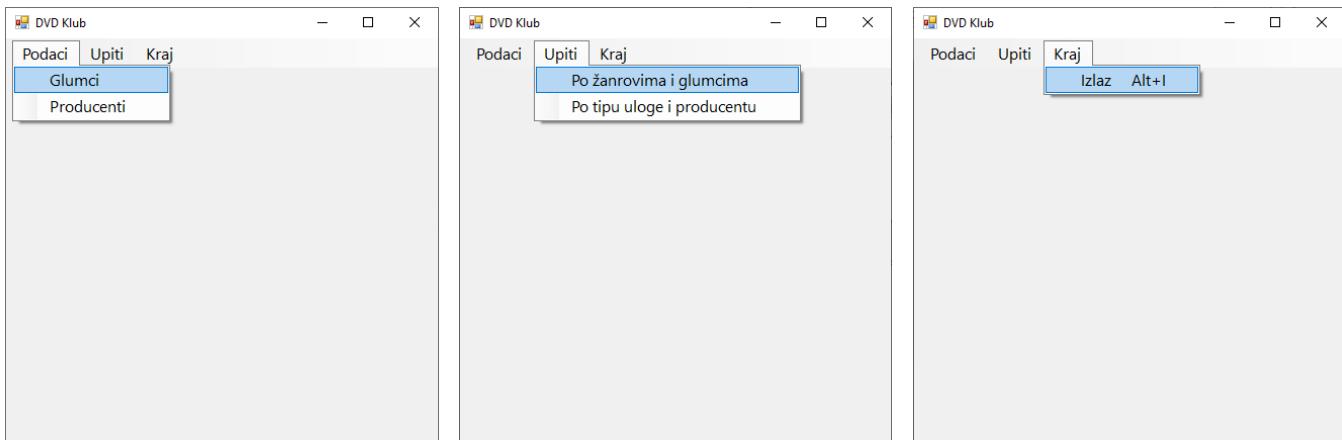


Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

За продуцента треба да се памти и адреса веб странице на којој врши промоцију својих филмова.

Награде за најбоље филмове се додељују од 1923 године. Филмови могу, али и не морају да добијају награде као најбољи филмови на различитим фестивалима. Један филм може да добије више награда. Ако је филм добио награду, памти се година добијања награде. Свака награда се карактерише, називом (Оскар) и годином од када се додељује (1929).

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о глумцима обавља се преко обрасца чији је изглед дат на слици:

Šifra	Ime	Prezime	Datum rođenja	Mesto rođenja
19	Matthew	Perry	24.01.1974	Cleveland
20	Sean	Penn	23.04.1948	Cleveland
21	John	Travolta	02.04.1965	Montgomery
22	Eric	Bana	24.09.1956	
23	Ingrid	Bergman	31.01.1938	Miami
24	Daniel	Day-Lewis	07.07.1974	Fort Worth
25	Arnold	Schwarzenegger	18.07.1969	

У оквиру са листом приказују се постојећи подаци о глумцима сортирани по шифри у растући низ.

Ако се унесе шифра која постоји, приказују се подаци о том глумцу.

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Po zanrovima i glumcima**, приказује називе филмова одређеног жанра, назив лика и врсту улоге, за траженог глумца који је глумио у том филму сортираних по времену трајања филма у опадајући низ.

Образац треба да изгледа као на слици:

Moderna vremena	Brad Schultz	Glavna uloga
Casablanca	Raquel Murillo	Epizodna uloga
Titanic	Megan Mathis	Epizodna uloga
Diplomac	Marisol Rangel	Sporedna uloga

Називи жанрова су сортирани у растући низ.

Резултати упита приказују се у оквиру са листом.

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б3

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б3
Назив радног задатка	ДВД КОЛЕКЦИЈА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Приказане су иницијалне вредности у оквиру са листом	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 41)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако унета шифра постоји	2	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Додата ставка у оквир са листом после уписивања новог слога у базу	2	0
Обезбеђено брисање података из базе по спецификацији задатка	9	0
Обрисана ставка из оквира са листом после брисања слога из базе	2	0
Обезбеђено ажурирање података у бази по спецификацији задатка	10	0
Измењена ставка у оквир са листом после ажурирања слога у бази	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за жанр	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Приказани подаци у оквир са листом	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

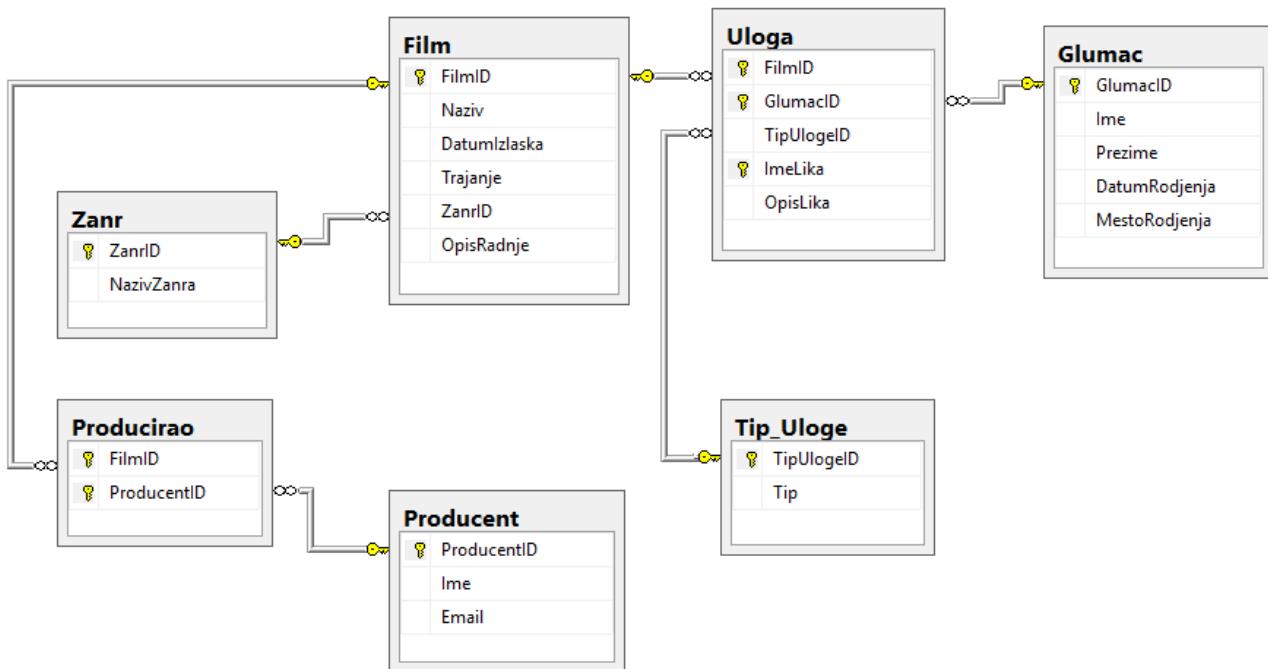
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б4

НАЗИВ РАДНОГ ЗАДАТКА: ДВД КОЛЕКЦИЈА

ЕИТ-Б4: ДВД КОЛЕКЦИЈА

Креирана је база података у којој се чувају подаци о твојој личној ДВД колекцији.

Дијаграм базе је дат на слици:

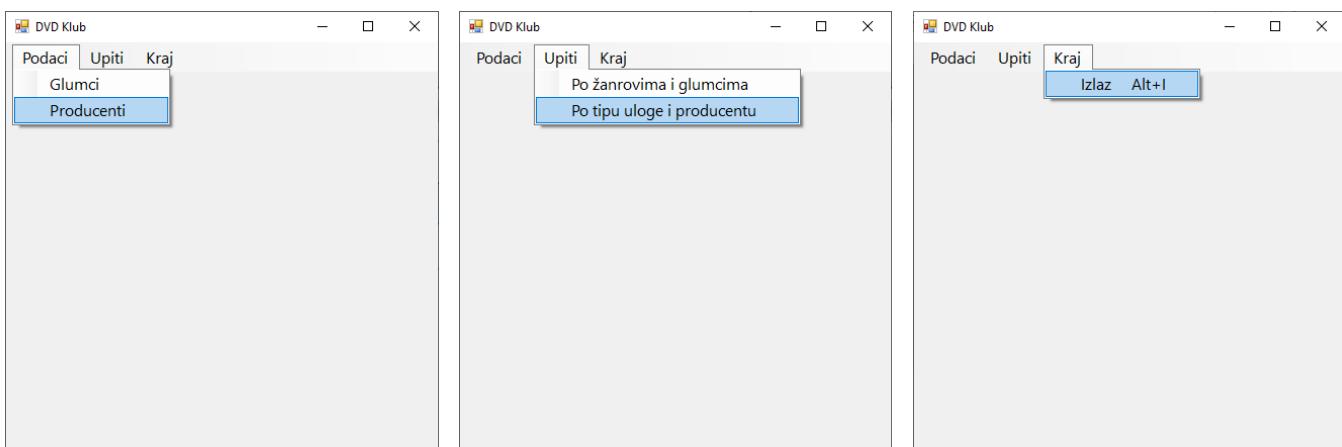


Користећи DDL наредбе, проширити базу је тако да се задовоље следећи захтеви:

За продуцента треба да се памти и адреса веб странице на којој врши промоцију својих филмова.

Награде за најбоље филмове се додељују од 1923 године. Филмови могу, али и не морају да добијају награде као најбољи филмови на различитим фестивалима. Један филм може да добије више награда. Ако је филм добио награду, памти се година добијања награде. Свака награда се карактерише, називом (Оскар) и годином од када се додељује (1929).

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о продуцентима обавља се преко обрасца чији је изглед дат на слици:

Šifra	Naziv producentske kuće	E-mail adresa
9	20th Century Fox	wjtk@outlo ^
17	American International Pictures	skrcpn72@h
8	BBC Films	gvymt90@gm
18	Capitol Films	pzqim67@ou
19	Cinemation Industries	bzmcn5@hot
20	Columbia TriStar Motion Picture Group	xfsojp18@o
6	DreamWorks	ummi7@mail v
4	Four Star Television	

У оквиру са листом приказују се постојећи подаци о продуцентима сортирани по шифри у растући низ.

Ако се унесе шифра која постоји, приказују се подаци о том продуценту.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Po vrsti uloge i producentu**, приказује назив филма, жанр, име и презиме глумца који су играли одређени тип улоге, у филмовима које је продуцирао тражени продуцент, сортиране по називу филма у растући низ.

Образац треба да изгледа као на слици:

Film	Filski žanr	Glumac
Avijaticar	Dramski film	John Goodman
Avijaticar	Dramski film	Ben Affleck
Carlijevi andeli	Akcijski	Omar Sharif
Carlijevi andeli	Akcijski	Sylvester Stallone
Elizabeth	Istorijski film	Helen Hunt
Elizabeth	Istorijski film	Ingrid Bergman
Hrabro srce	Triler	Nicolas Cage
Hrabro srce	Triler	Kevin Costner

Називи типа улоге сортирани су у растућем низу.

Називи продуцента сортирани су у растућем низу.

Резултати упита приказују се у оквиру са листом.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б4

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б4
Назив радног задатка	ДВД КОЛЕКЦИЈА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност у оквир са листом	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 36)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако унета шифра постоји	2	0
Обезбеђено уписивање података у базу по спецификацији задатка	8	0
Додата ставка у оквиру са листом после уписа новог слога у базу	2	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Обрисана ставка у оквиру са листом после брисања слога из базе	2	0
Обезбеђено ажурирање података у бази по спецификацији задатка	8	0
Измењена ставка у оквиру са листом после ажурирања слога у бази	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 14)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за тип улоге	5	0
Постављена иницијална вредност за производника	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Приказани подаци у оквиру са листом	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

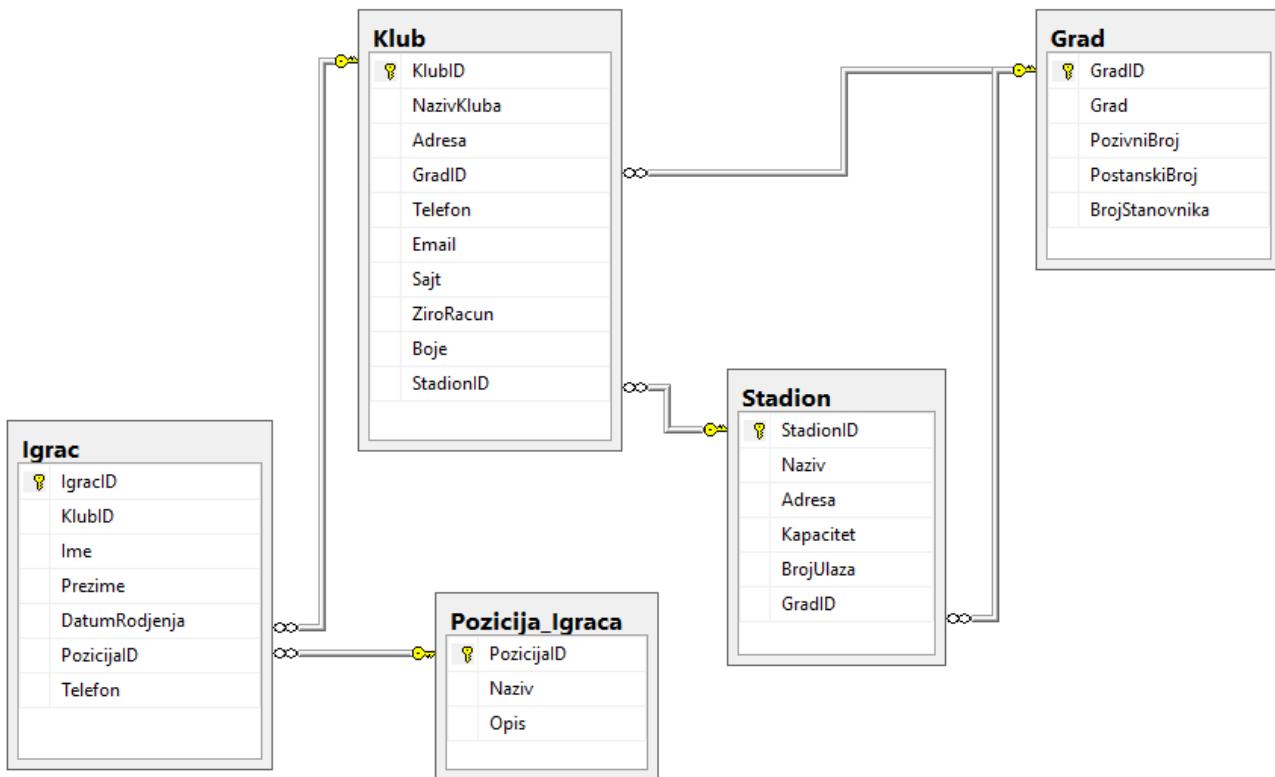
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б5

НАЗИВ РАДНОГ ЗАДАТКА: ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ

ЕИТ-Б5: ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ

Креирана је база података у којој ће се водити евиденција о клубовима у Фудбалској лиги Србије.

Дијаграм базе је дат на слици:



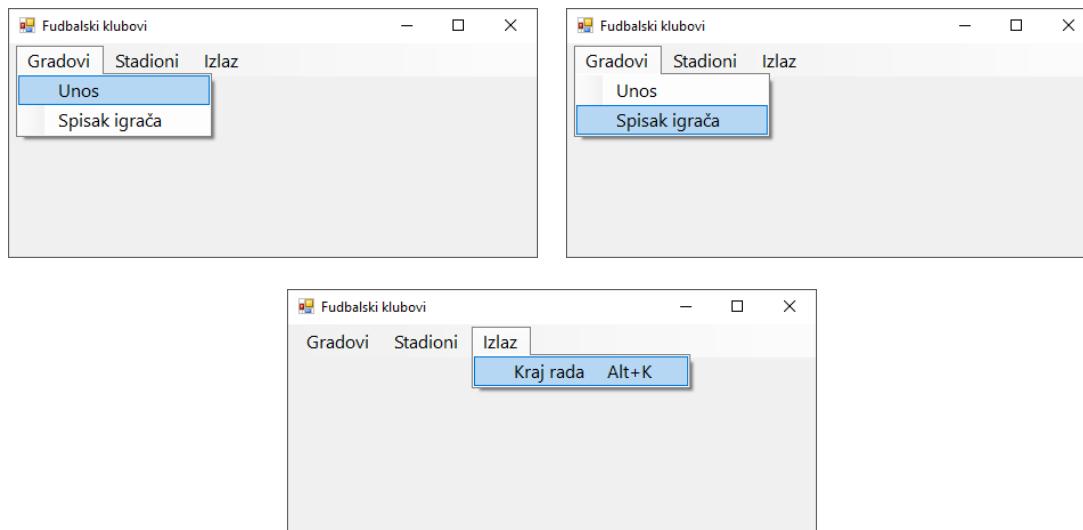
Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

За сваког играча се води статистика о учинку на утакмицама: број постигнутих голова, број црвених и жутих картона, број фаула које је направио и број фаула који су над њим направљени. (Наравно – на једној утакмици играју више играча, а један играч може да игра на више утакмица).

За сваку утакмицу се води евиденција који је клуб био домаћин на утакмици, који клуб је био гост, ког датума је утакмица одиграна, у које време и за које такмичење се игра.

Такмичење осим свог идентификационог броја има и свој назив (Лига шампиона).

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о градовима обавља се преко обрасца чији је изглед дат на слици:

Šifra

Grad

Pozivni broj

Poštanski broj

Broj stanovnika

Prethodni Sledi Novi Upiši

Izadji

Приликом отварања обрасца приказује се први слог.

Притиском на дугме **Novi** ресетују се сва поља и курсор се поставља на поље за уношење шифре.

Ако је поље за шифру празно, притиском на дугме **Prethodni** приказује се први слог.

Ако је поље за шифру празно притиском на дугме **Sledeći** приказује се последњи слог.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису података, као и о обрађеним грешкама.

Ставка менија **Spisak igraca**, приказује имена, презимена и називе клубова у којима играју играчи који долазе из траженог града и који играју на одређеним позицијама, сортирани по имену и презимену у растући низ, а затим по називу клуба у опадајући низ.

Образац треба да изгледа као на слици:

Grad Prikaži Izadji

Pozicija

	Ime	Prezime	Klub	Grad	Naziv
▶	Balša	Dačić	FK Duboči...	Leskovac	Igrač sredi...
	Matija	Hajdeman	FK Duboči...	Leskovac	Igrač sredi...
	Strahinja	Daničić	FK Duboči...	Leskovac	Igrač sredi...
*					

За град се приказују називи сортирани у растући низ.

Резултати упита се приказују у контроли за табеларни приказ података.

Дугме **Izadjii** затвара отворе обрасца, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б5

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б5				
Назив радног задатка	ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Приказан први слог табеле	2	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 36)	ТАЧНО	НЕТАЧНО
Обезбеђена функционалност дугмета за унос новог слога	3	0
Обезбеђено уписивање података у базу по спецификацији задатка	11	0
Обезбеђена функционалност дугмета за приказ следећег слога	9	0
Обезбеђена функционалност дугмета за приказ претходног слога	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 11)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	3	0
Постављена иницијална вредност за град	4	0
Постављена иницијална вредност за позицију играча	4	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

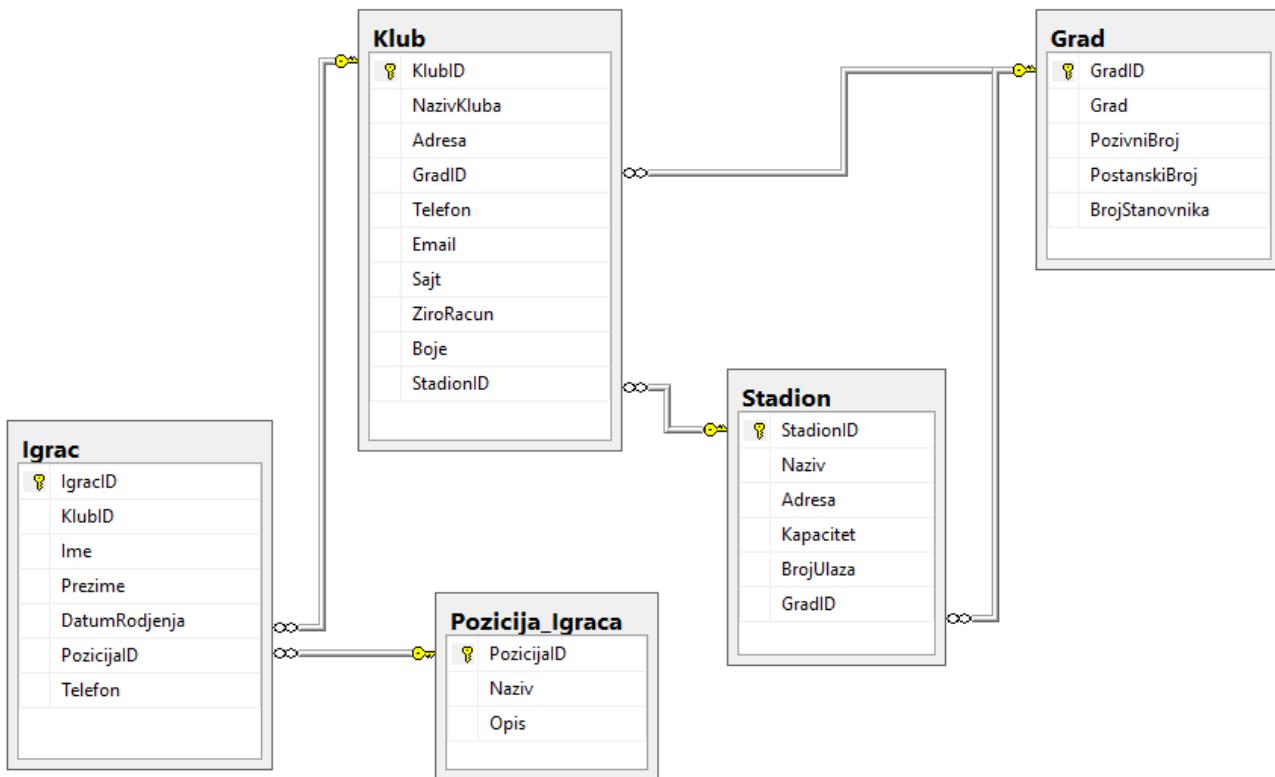
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б6

НАЗИВ РАДНОГ ЗАДАТКА: ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ

ЕИТ-Б6: ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ

Креирана је база података у којој ће се водити евиденција о клубовима у Фудбалској лиги Србије.

Дијаграм базе је дат на слици:



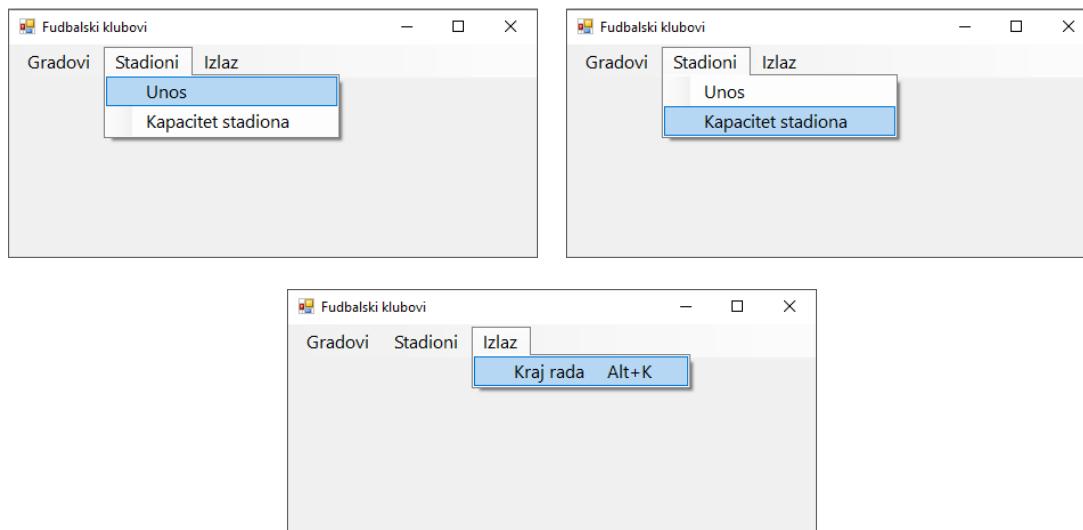
Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

За сваког играча се води статистика о учинку на утакмицама: број постигнутих голова, број црвених и жутих картона, број фаула које је направио и број фаула који су над њим направљени. (Наравно – на једној утакмици играју више играча, а један играч може да игра на више утакмица).

За сваку утакмицу се води евиденција који је клуб био домаћин на утакмици, који клуб је био гост, ког датума је утакмица одиграна, у које време и за које такмичење се игра.

Такмичење осим свог идентификационог броја има и свој назив (Лига шампиона).

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о стадионима обавља се преко обрасца чији је изглед дат на слици:

Šifra 1
Naziv Stadion Pamučnog kombinata Jumko
Adresa Jovana Cvijića 23
Kapacitet 14000
Broj ulaza 12
Grad Pančevo

Prethodni Sledeći Novi Upiši
Izadji

За град се приказују називи градова сортирани у растући низ.

Приликом отварања обрасца приказује се први слог.

Притиском на дугме **Novi** ресетују се сва поља и курсор се поставља на поље за уношење шифре.

Ако је поље за шифру празно, притиском на дугме **Prethodni** приказује се први слог.

Ако је поље за шифру празно притиском на дугме **Sledeći** приказује се последњи слог.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису података, као и о обрађеним грешкама.

Ставка менија **Kapacitet stadiona**, приказује листу назива свих стадиона који имају већи капацитет од траженог, назив клуба који на њему тренира, веб страницу клуба, капацитет и број улаза стадиона, сортираних по капацитету у растући низ, а по броју улаза у опадајући низ.

Образац треба да изгледа као на слици:

	Naziv	Klub	Sajt	Kapacitet	
▶	Stadion Badnjevac	FK FAP	www.nuxf...	22000	1
	Stadion Partizana	FK Radnič...	www.mrmz...	24000	7
	Stadion Bojan Majić	FK Metal...	www.fbzop...	26000	1
	Stadion Borca	FK BASK	www.yxyo...	27000	1
	Stadion Careva ču...	FK Mladost...	www.lkqse...	28000	1
	Stadion Promena...	FK Bežanija	www.udpa....	28000	1

Резултати упита приказују се у контроли за табеларни приказ података.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б6

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б6
Назив радног задатка	ФУДБАЛСКИ КЛУБ И УТАКМИЦЕ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 14)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Приказан први слог табеле	2	0
Постављена иницијална вредност за називе	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 38)	ТАЧНО	НЕТАЧНО
Обезбеђена функционалност дугмета за унос новог слога	3	0
Обезбеђено уписивање података у базу по спецификацији задатка	11	0
Обезбеђена функционалност дугмета за приказ следећег слога	9	0
Обезбеђена функционалност дугмета за приказ претходног слога	9	0
Проверена вредност која се уноси на поље за капацитет	2	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 4)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

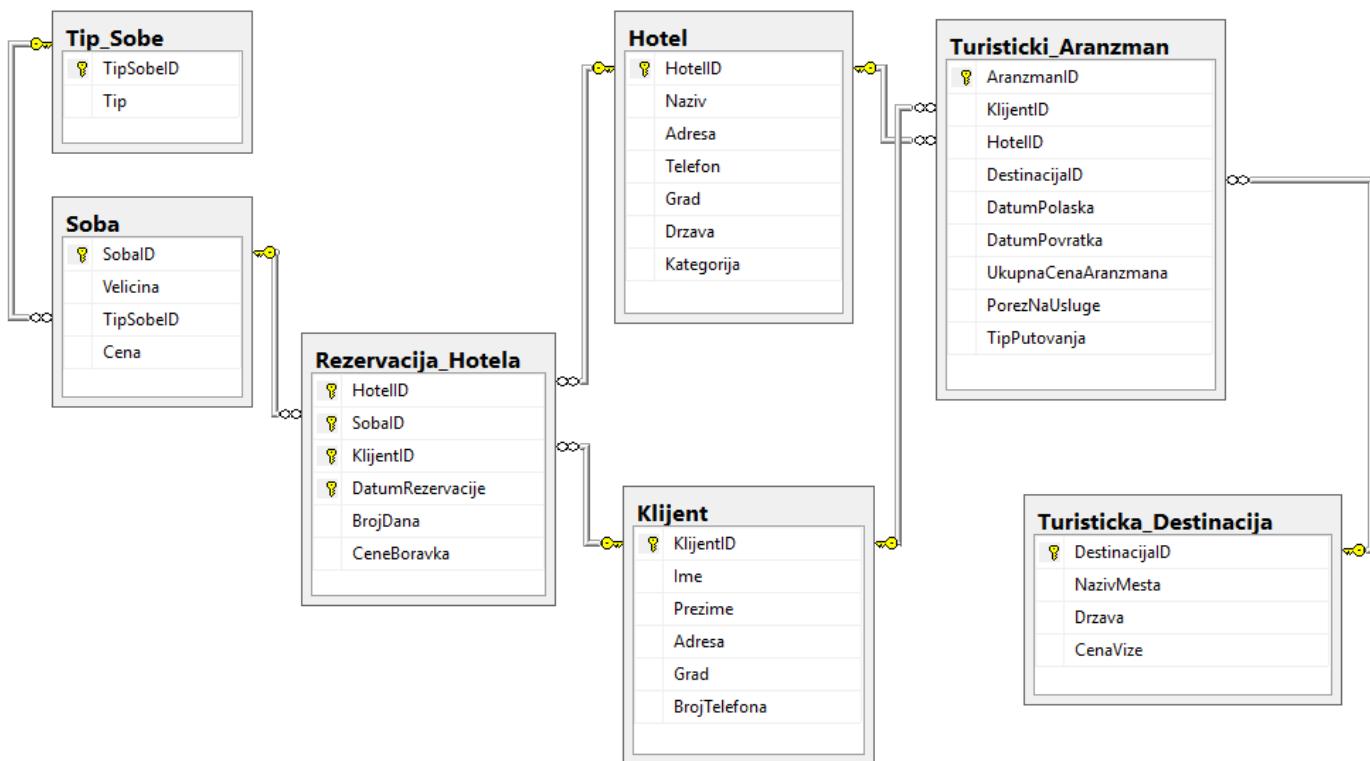
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б7

НАЗИВ РАДНОГ ЗАДАТКА: ТУРИСТИЧКА АГЕНЦИЈА

ЕИТ-Б7: ТУРИСТИЧКА АГЕНЦИЈА

Креирана је база података у којој се чувају подаци о туристичким аранжманима једне агенције:

Дијаграм базе је дат на слици:

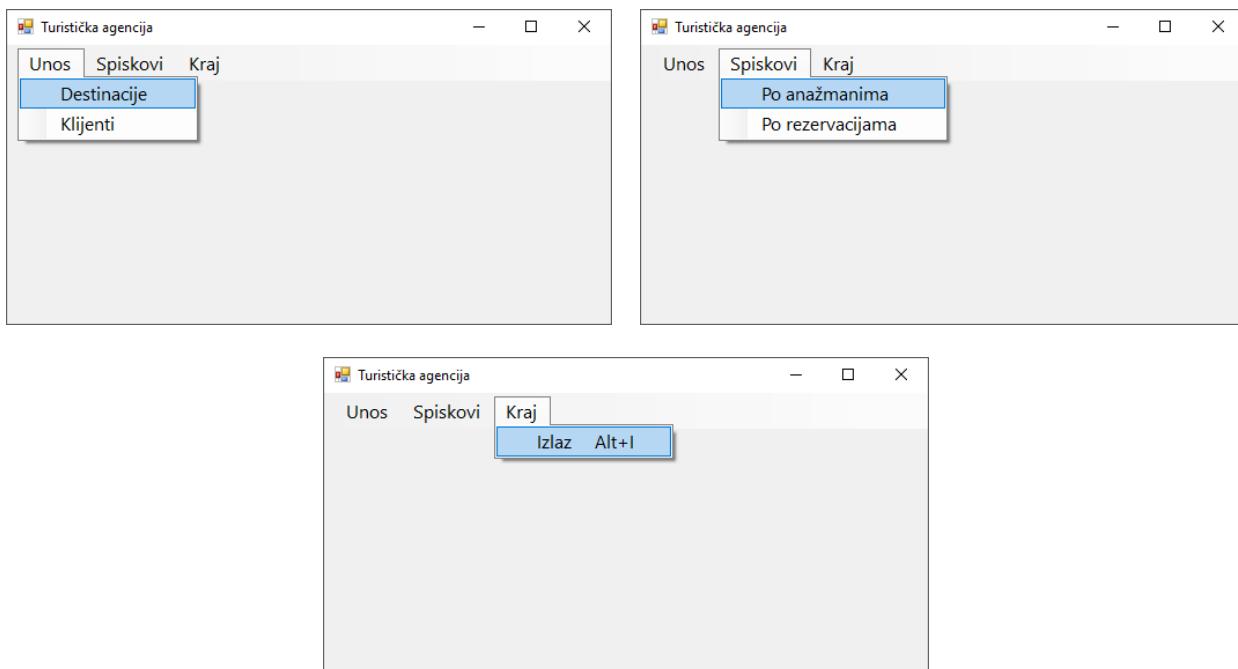


Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

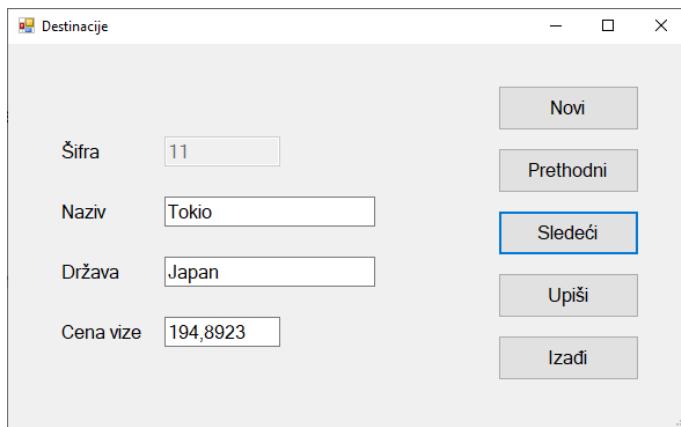
Клијентима смо у могућности да понудимо додатне обиласке да би учинили наш аранжман атрактивнијим.

За сваки обилазак памти се назив обиласка и цена. Повезати ову табелу са табелом Туристички аранжман.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о дестинацијама обавља се преко обрасца чији је изглед дат на слици:



Приликом отварања обрасца приказује се први слог.

Притиском на дугме **Novi** ресетирују се сва поља и курсор се поставља на поље за уношење шифре.

Ако је поље за шифру празно притиском на дугме **Prethodni** приказује се први слог.

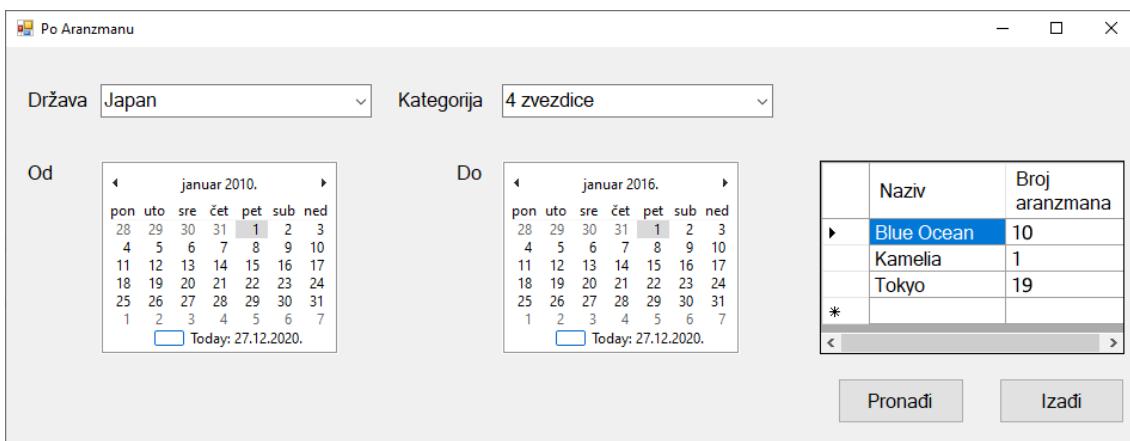
Ако је поље за шифру празно притиском на дугме **Sledeći** приказује се последњи слог.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије информацију о успешном уписивању у табелу, као и о обрађеним грешкама.

Ставка менија **Po aranzmanima**, приказује списак назива хотела одређене категорије и из одређене државе и укупан број аранжмана за сваки хотел, који су обављени у траженом периоду, сортиране по називу хотела у растући низ.

Образац треба да изгледа као на слици:



За категорију се приказују називи сортирани у растући низ.

За државу се приказују називи сортирани у растући низ.

Резултат упита се приказује у контроли за табеларни приказ података.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима

За оцењивање користити образац за оцењивање број ЕИТ – Б7

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б7
Назив радног задатка	ТУРИСТИЧКА АГЕНЦИЈА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Проширена табела 2	3	0
Дефинисани кључеви у табели 2	4	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Приказан први слог табеле	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 36)	ТАЧНО	НЕТАЧНО
Обезбеђена функционалност дугмета за унос новог слога	3	0
Обезбеђено уписивање података у базу по спецификацији задатка	11	0
Обезбеђена функционалност дугмета за приказ следећег слога	9	0
Обезбеђена функционалност дугмета за приказ претходног слога	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 16)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за категорију	6	0
Постављена иницијална вредност за државу	6	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	17	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

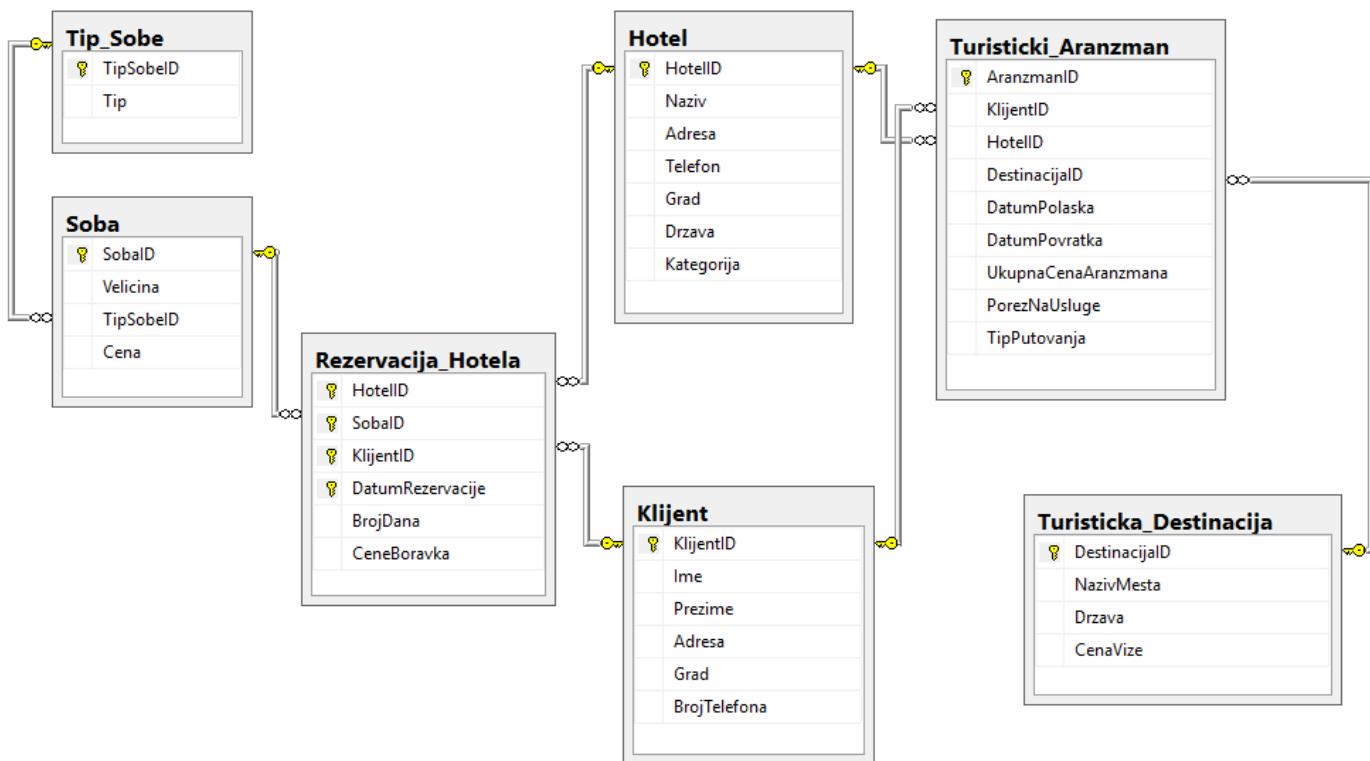
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б8

НАЗИВ РАДНОГ ЗАДАТКА: ТУРИСТИЧКА АГЕНЦИЈА

ЕИТ-Б8: ТУРИСТИЧКА АГЕНЦИЈА

Креирана је база података у којој се чувају подаци о туристичким аранжманима једне агенције:

Дијаграм базе је дат на слици:

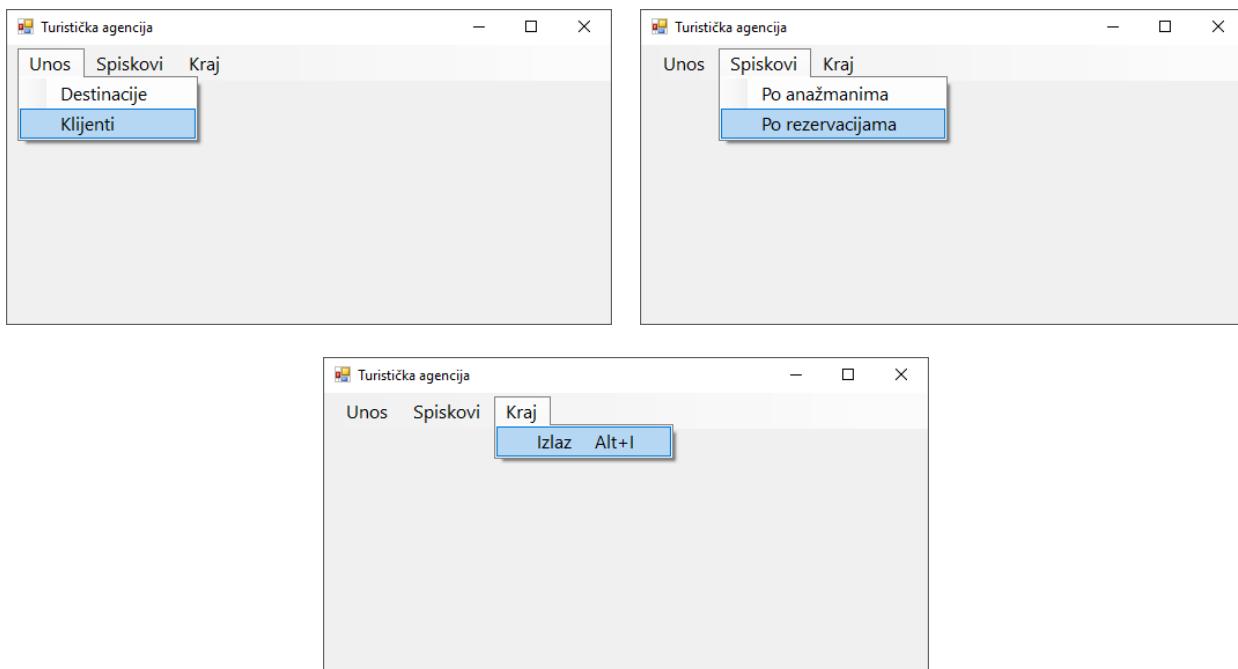


Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

Клијентима смо у могућности да понудимо додатне обиласке да би учинили наш аранжман атрактивнијим.

За сваки обилазак памти се назив обиласка и цена. Повезати ову табелу са табелом Туристички аранжман.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о клијентима обавља се преко обрасца чији је изглед дат на слици:

Приликом отварања обрасца приказује се први слог.

Притиском на дугме **Novi** ресетују се сва поља и курсор се поставља на поље за уношење шифре.

Ако је поље за шифру празно, притиском на дугме **Prethodni** приказује се први слог.

Ако је поље за шифру празно, притиском на дугме **Sledeći** приказује се последњи слог.

Дугме **Izadjii** затвара отворе образац, и враћа на почетни мени.

Обезбедити да корисник добије информацију о успешном уписивању у табелу, као и о обрађеним грешкама.

Ставка менија **Po rezervacijama**, приказује списак назива хотела одређене категорије и из одређене државе, и укупан број резервација за сваки хотел, који су обављени у траженом периоду, сортиране по називу хотела у растући низ.

Образац треба да изгледа као на слици:

За категорију се приказују називи сортирани у растући низ.

За државу се приказују називи сортирани у растући низ.

Резултат упита се приказује у контроли за табеларни приказ података.

Дугме **Izadjii** затвара отворе образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б8

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б8
Назив радног задатка	ТУРИСТИЧКА АГЕНЦИЈА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Проширена табела 2	3	0
Дефинисани кључеви у табели 2	4	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Приказан први слог табеле	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 36)	ТАЧНО	НЕТАЧНО
Обезбеђена функционалност дугмета за унос новог слога	3	0
Обезбеђено уписивање података у базу по спецификацији задатка	11	0
Обезбеђена функционалност дугмета за приказ следећег слога	9	0
Обезбеђена функционалност дугмета за приказ претходног слога	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 16)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за категорију	6	0
Постављена иницијална вредност за државу	6	0

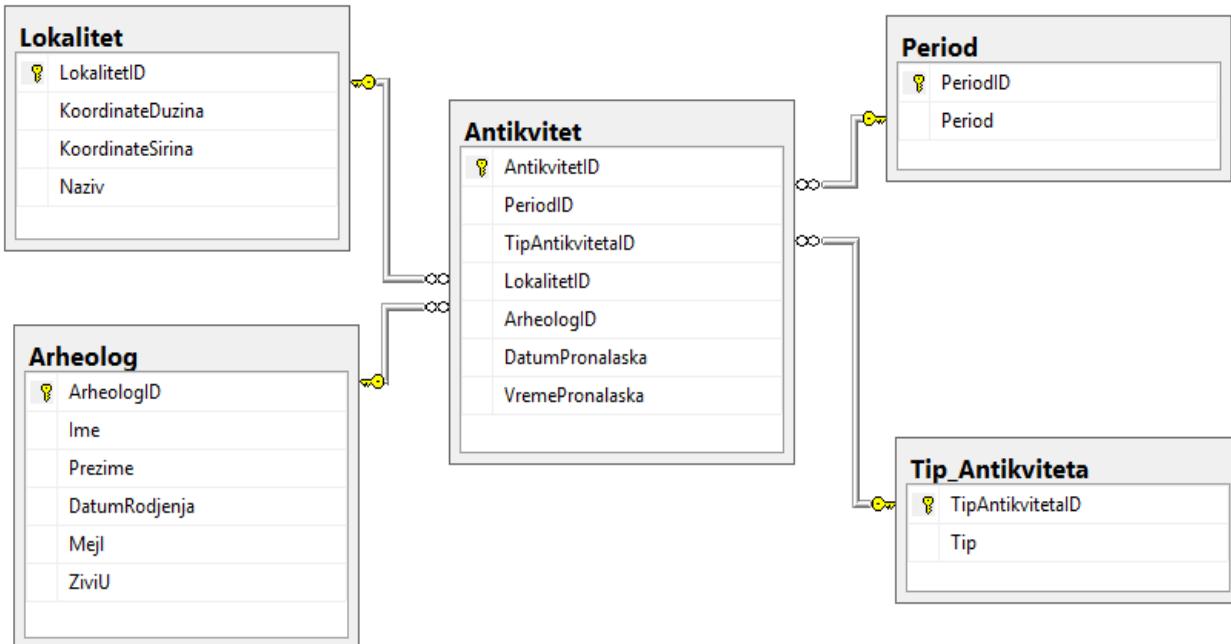
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	17	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б9: АНТИКВИТЕТИ И ЛОКАЦИЈЕ

За потребе археолога који истражује антиквите на различитим локалитетима у Србији и из различитих периода креирана је база података на основу којих се утврђује како се Србија мењала кроз историју.

Дијаграм базе је дат на слици:

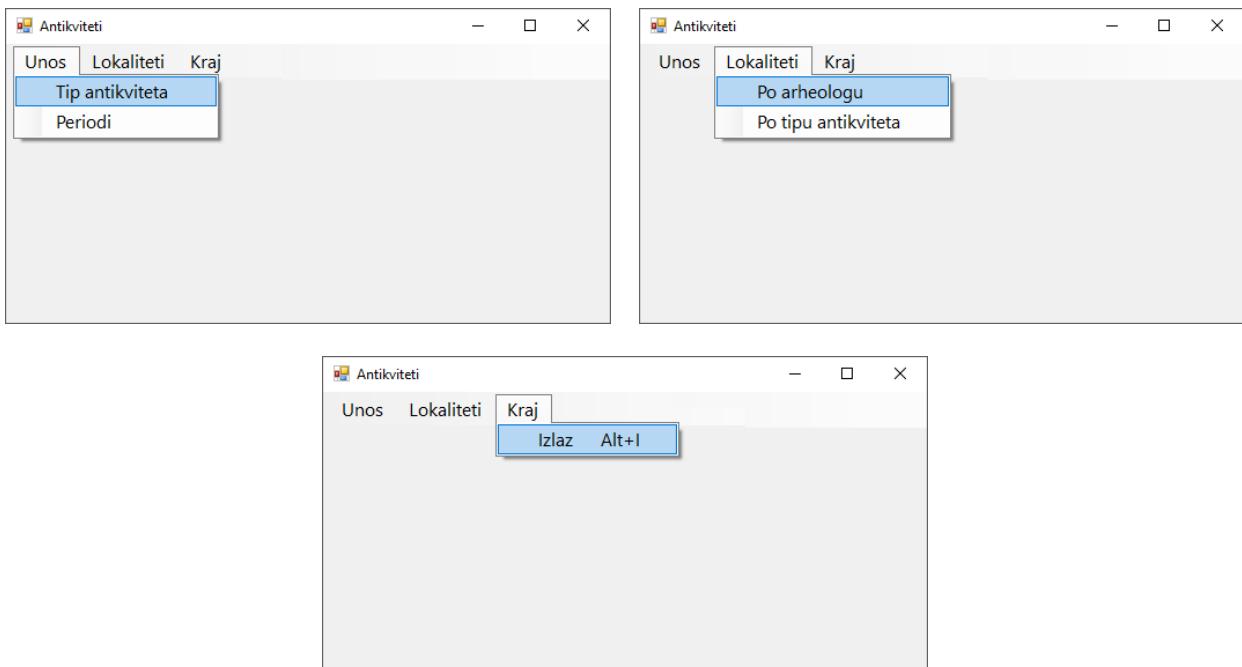


Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

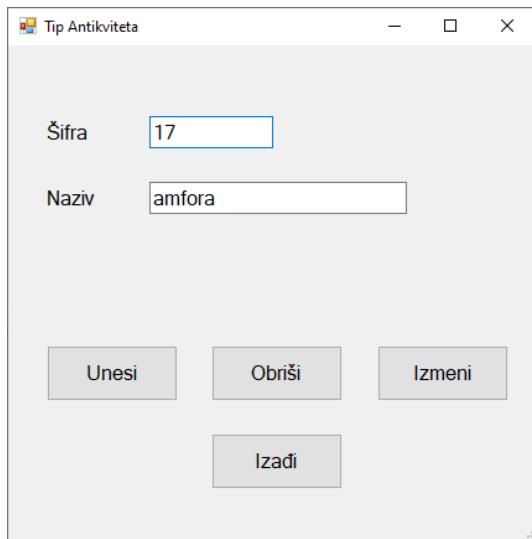
Локалитети се налазе близу градова. Потребно је креирати табелу у којој би се чували подаци о имену града, његовом позивном броју, поштанском броју и броју становника.

Обзиром да су се границе Србије мењале кроз историју и да и ван граница наше земље можемо наћи антиквите везане за наш народ, потребно је чувати податке и о држави у којој се град налази, као и позивни број за државу и њен број становника.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о типу антиквитета обавља се преко обрасца чији је изглед дат на слици:



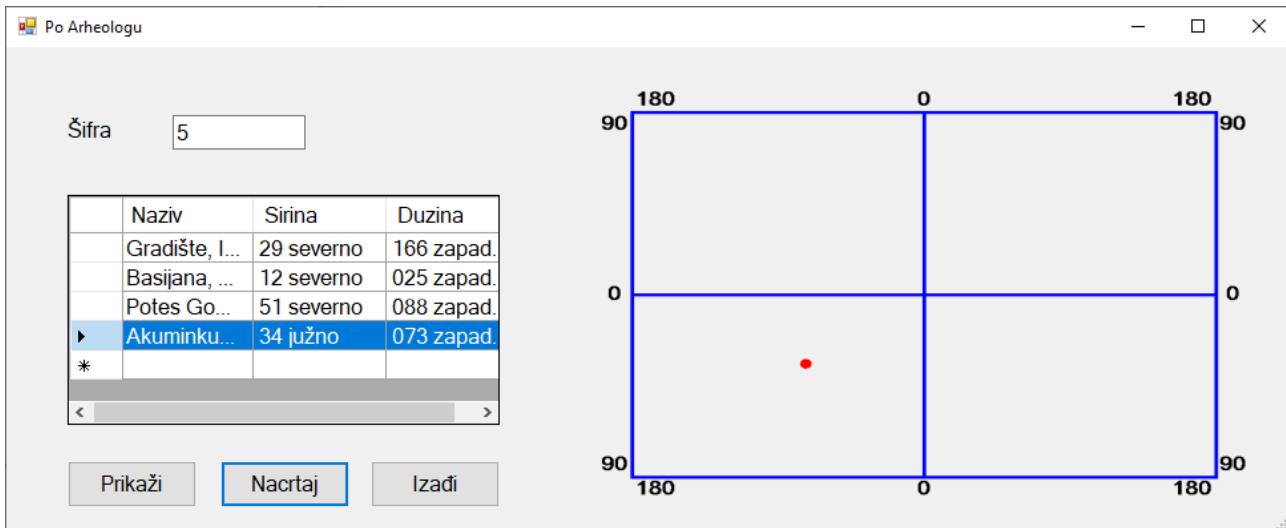
Уколико унета шифра типа антиквитета постоји приказати његов назив.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Po arheologu**, приказује називе локалитета, координате његове географске дужине и ширине, за оне локалитете у чијем истраживању је учествовао тражени археолог.

Образац треба да изгледа као на слици:



Подаци се приказују у контроли за табеларни приказ података.

За изабрани локалитет из контроле за табеларни приказ података, графички се приказује географски положај локалитета.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ– Б9

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б9
Назив радног задатка	АНТИКВИТЕТИ И ЛОКАЦИЈЕ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Проширена табела	2	0
Дефинисан кључ у проширеној табели	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 7)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако унета шифра постоји	2	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Обезбеђено брисање података из базе по спецификацији задатка	8	0
Обезбеђена измена података у бази по спецификацији задатка	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Нацртан правоугаоник	8	0

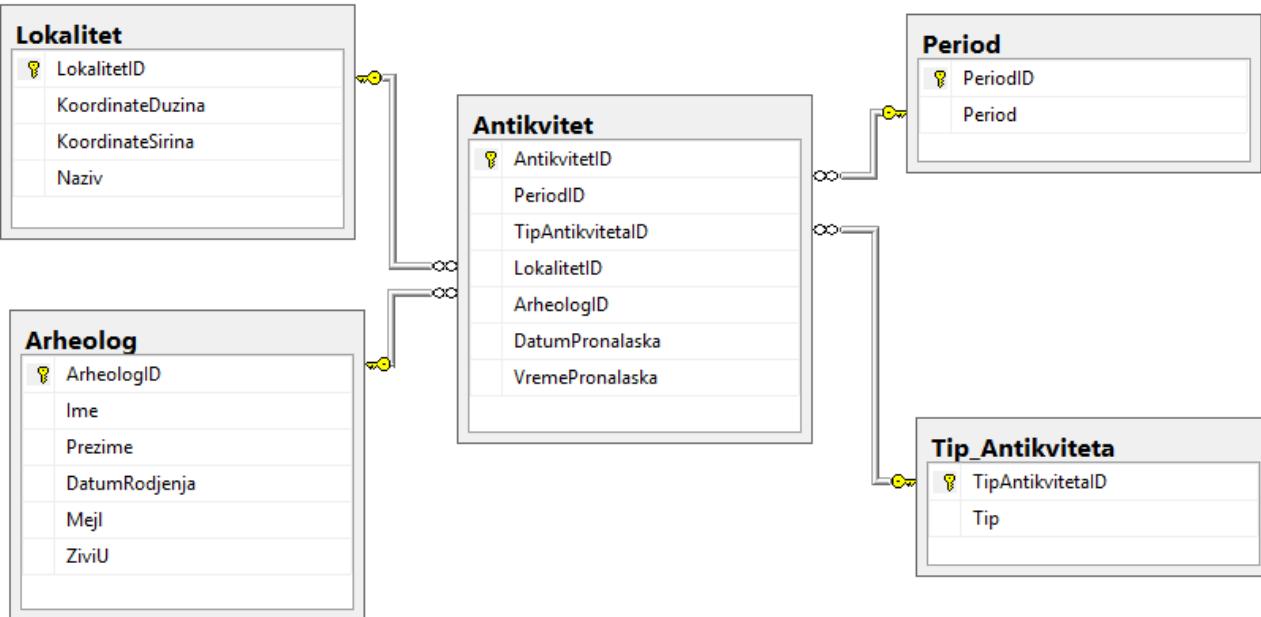
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 28)	ТАЧНО	НЕТАЧНО
Приказани подаци у контроли за табеларни приказ података	12	0
Графички приказан положај изабраног локалитета	12	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б10: АНТИКВИТЕТИ И ЛОКАЦИЈЕ

За потребе археолога који истражује антиквитетете на различитим локалитетима у Србији и из различитих периода креирана је база података на основу којих се утврђује како се Србија мењала кроз историју.

Дијаграм базе је дат на слици:

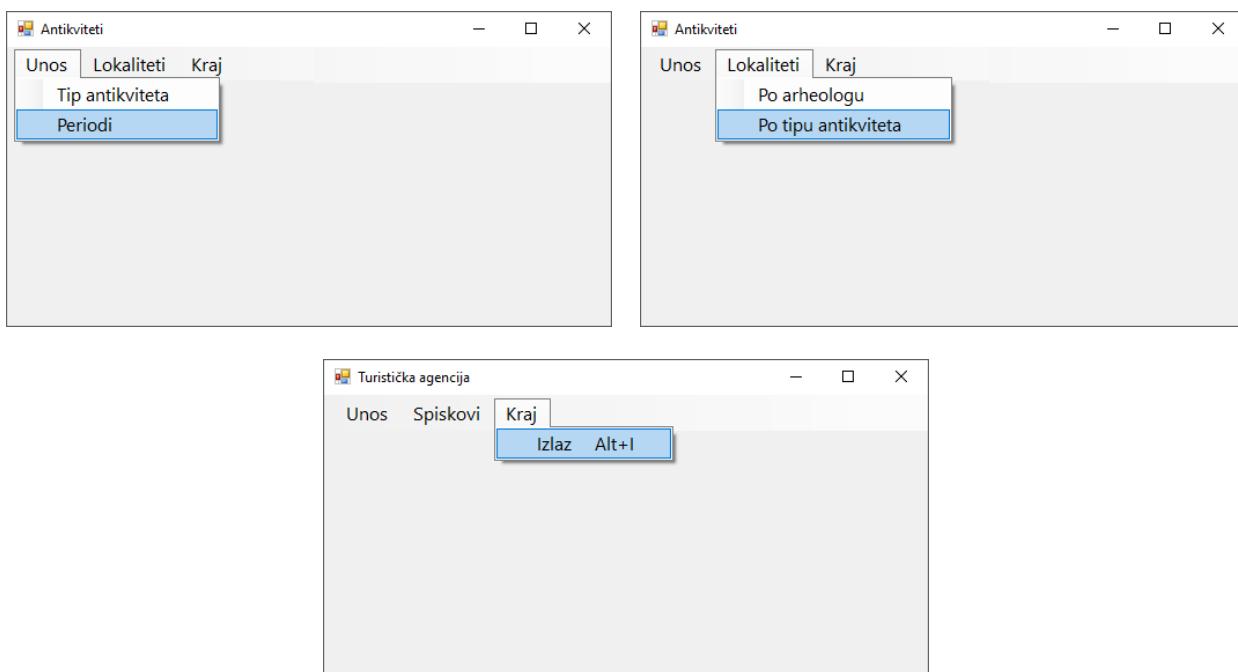


Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

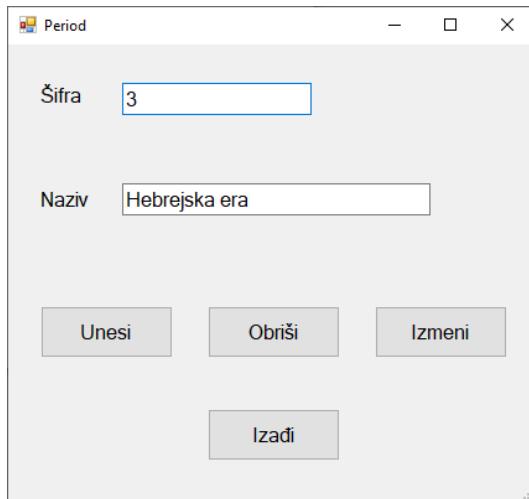
Локалитети се налазе близу градова. Потребно је креирати табелу у којој би се чували подаци о имену града, његовом позивном броју, поштанском броју и броју становника.

Обзиром да су се границе Србије мењале кроз историју и да и ван граница наше земље можемо наћи антиквите везане за наш народ, потребно је чувати податке и о држави у којој се град налази, као и позивни број за државу и њен број становника.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о периодима обавља се преко обрасца **Periodi**, чији је изглед дат на слици:



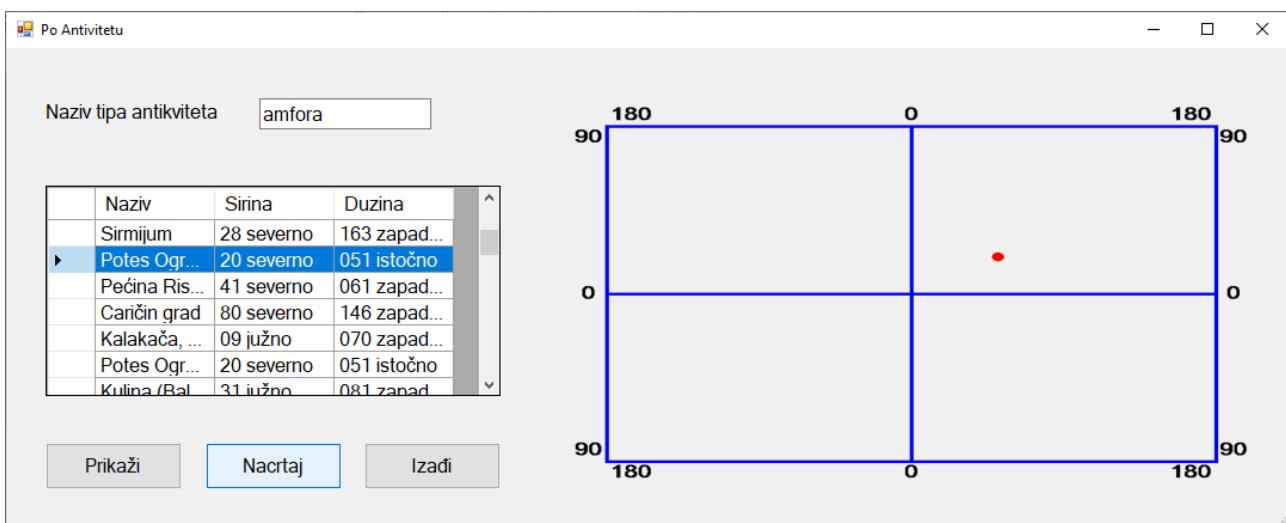
Уколико унета шифра периода постоји приказати његов назив.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Po tipu antikviteta**, приказује шифре назива локалитета, координате његове географске дужине и ширине, за оне локалитете на којима је пронађен одређени антиквитет.

Образац треба да изгледа као на слици:



Подаци се приказују у контроли за табеларни приказ података.

За изабрани локалитет из контроле за табеларни приказ података, графички се приказује географски положај локалитета.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ-Б10

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б10
Назив радног задатка	АНТИКВИТЕТИ И ЛОКАЦИЈЕ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Проширена табела	2	0
Дефинисан кључ у проширеној табели	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 7)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако унета шифра постоји	2	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Обезбеђено брисање података из базе по спецификацији задатка	8	0
Обезбеђена измена података у базу по спецификацији задатка	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Нацртан правоугаоник	8	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 28)	ТАЧНО	НЕТАЧНО
Приказани подаци у контроли за табеларни приказ података	12	0
Графички приказан положај изабраног локалита	12	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

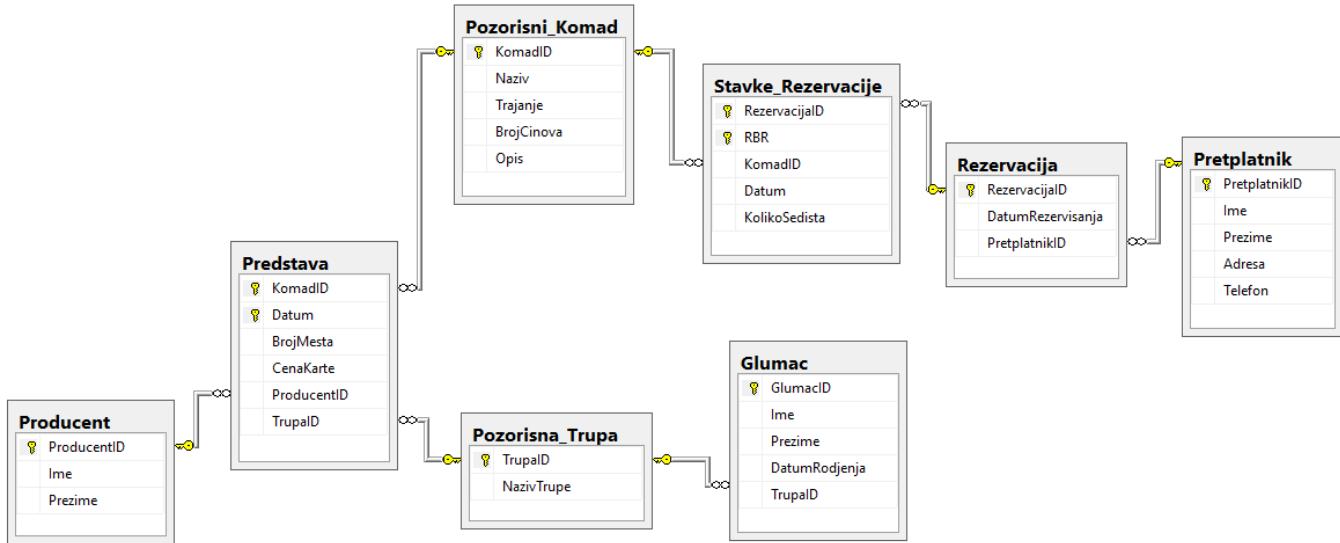
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б11

НАЗИВ РАДНОГ ЗАДАТКА: ПОЗОРИШНЕ ПРЕДСТАВЕ

ЕИТ-Б11: ПОЗОРИШНЕ ПРЕДСТАВЕ

За потребе позоришта креирана је база података о позоришним представама.

Дијаграм за постојећу базу дат је на слици:



Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

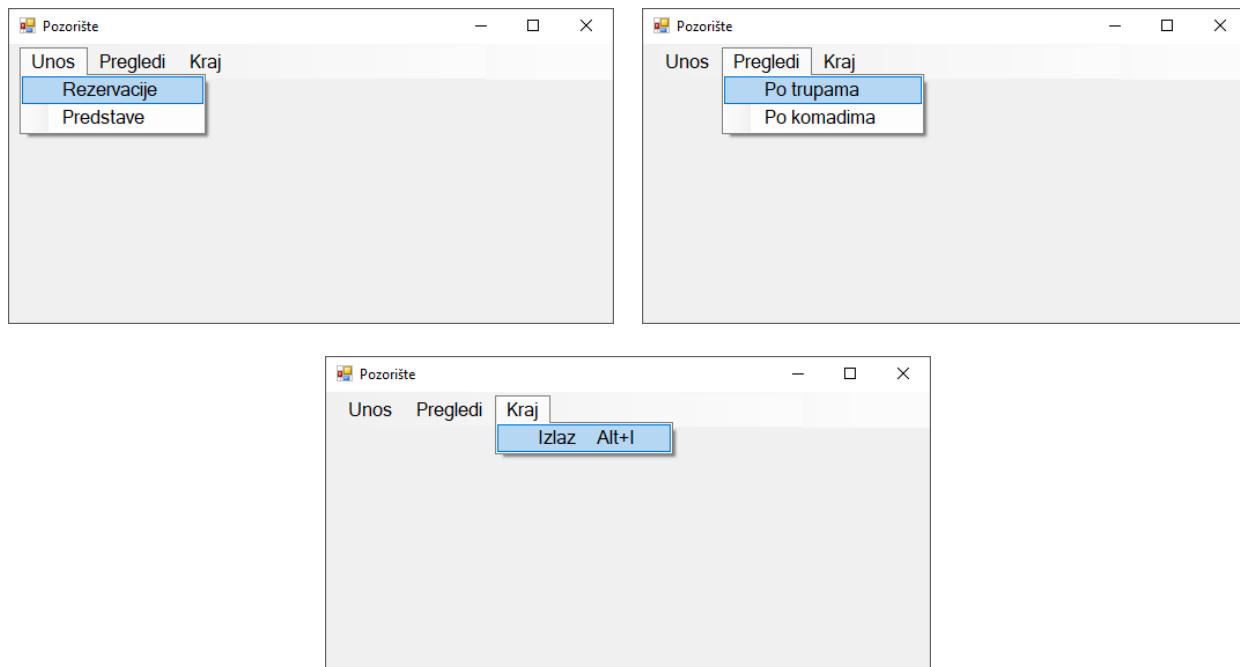
У циљу побољшања услуга и осавремењивању начина плаћања желимо да омогућимо претплатницима да резервишу и плаћају изврше путем картица. Због тога је потребно проширити базу са следећим параметрима:

Креирати табелу картица у којој ће се чувати број картице и датум важења.

Креирати табелу тип картице у којој ће се памтити подаци о типу картице и кратак опис.

Табелу картице повезати са табелом претплатник при чему у табели претплатник додати нову колону број картице.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о резервацијама обавља се преко обрасца чији је изглед дат на слици:

За комад се приказује шифра комада и назив комада сортирани у растући низ.

Уколико унета шифра резервације постоји приказује се датум резервације и име претплатника који је извршио резервацију.

Редни број ставке за сваку нову резервацију аутоматски се повећава за један код сваког новог уписа.

Могуће је обрисати само последњу ставку за одређену резервацију.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Po trupama**, приказује листу шифара, имена и презимена сортирану у растући абецедни низ по шифри свих глумаца који припадају одређеној трупи.

Образац треба да изгледа као на слици:

Glumac...	Ime	Prezime
11	Nemanja	Drenovac
28	Ibrahim	Filipović
33	Ivana	Aleksić
62	Uroš	Erceg
69	Hana	Andrejić
79	Sanja	Drenovac
100	Vuk	Andrejić
103	Aleksandra	Barjaktarević
131	Stefan	Frdelian

Komad	Trajanje/min	Odigрано пута
Bure barata	2	
Grejalica	75	1
Iza kulisa		1
Jadnici		1
Kralj Ibi	90	1
Lažni car Šćepa...		2
Mačka na usijan...	85	5
Prva braćna noć	75	2
Sablja Dimiskije		2
Sumnjiwo lice		1

За трупу се приказују називи сортирани у растући низ.

Подаци се приказују уоквиру са листом.

У контроли за табеларни приказ података приказује се листа назива комада, трајања комада у минутима и број одиграних представа комада које је извела одабрана трупа.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ-Б11

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б11				
Назив радног задатка	ПОЗОРИШНЕ ПРЕДСТАВЕ				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						100

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Проширена табела	2	0
Дефинисан кључ у проширену табели	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за позоришне комаде	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 34)	ТАЧНО	НЕТАЧНО
Приказани подаци о претплатнику	2	0
Приказан редни број ставке за одабрану резервацију уколико је изабрана акција за упис	4	0
Приказан редни број ставке за одабрану резервацију уколико је изабрана акција за брисање	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	10	0
Обезбеђено брисање података из базе по спецификацији задатка	10	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за трупу	5	0

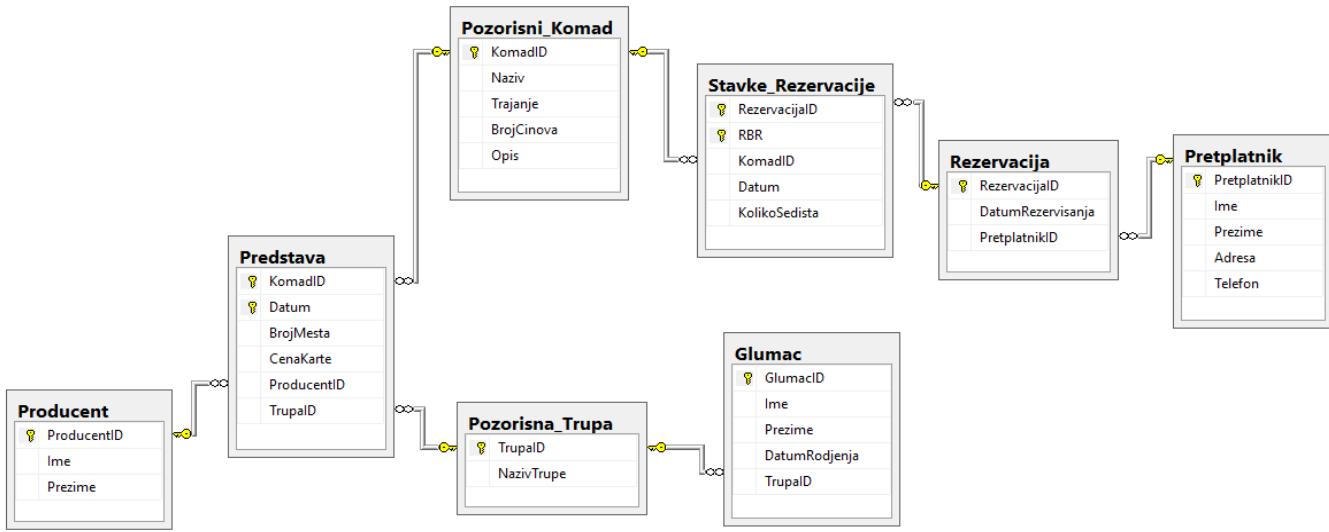
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Приказани подаци у оквиру са листом	8	0
Приказани подаци у контроли за табеларни приказ података	12	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б12: ПОЗОРИШНЕ ПРЕДСТАВЕ

За потребе позоришта креирана је база података о позоришним представама.

Дијаграм за постојећу базу дат је на слици:



Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

У циљу побољшања услуга и осавремењивању начина плаћања желимо да омогућимо претплатницима да резервишу и плаћање изврше путем картица. Због тога је потребно проширити базу са следећим параметрима:

Креирати табелу картица у којој ће се чувати број картице и датум важења.

Креирати табелу тип картице у којој ће се памтити подаци о типу картице и кратак опис.

Табелу картице повезати са табелом претплатник при чему у табели претплатник додати нову колону број картице.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:

Уношење података о представама обавља се преко обрасца чији је изглед дат на слици:

За комад се приказује шифра и назив комада сортирани у растући низ.

Ако није унета шифра позоришног комада, а датум јесте, обрисаће се све представе које су одигране унетог датума. Ако није унет датум, а шифра позоришног комада јесте, обрисаће се све представе унетог позоришног комада.

Ако није унета или шифра позоришне представе или датум, кориснику обавестити да није унео неки од ових података, и од њега тражи дозволу да обави брисање. Тек када корисник одобри брисање подаци се бришу.

Не дозволити брисање уколико није унета ни шифра позоришне представе ни датум.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Po komadima**, приказује листу назива и прихода које су оствариле позоришне трупе које су играле одређени комад у задатом временском интервалу.

Образац треба да изгледа као на слици:

По отварању обрасца у контролиза табеларни приказ података приказују се подаци о унетим позоришним представама.

Резултати упита приказују се у оквиру са листом.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б12

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б12				
Назив радног задатка	ПОЗОРИШНЕ ПРЕДСТАВЕ				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Проширена табела	2	0
Дефинисан кључ у проширеној табели над додатим пољем	3	0

Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	3	0
Постављена иницијална вредност за позоришни комад	4	0
Постављена иницијална вредност за производника	4	0
Постављена иницијална вредност за позоришну трупу	4	0

Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 33)	ТАЧНО	НЕТАЧНО
Обезбеђено уписивање података у базу по спецификацији задатка	12	0
Обезбеђено брисање података из базе по спецификацији задатка	11	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	3	0
Постављена иницијална вредност за позоришне комаде	4	0
Постављена иницијална вредност за датуме	2	0

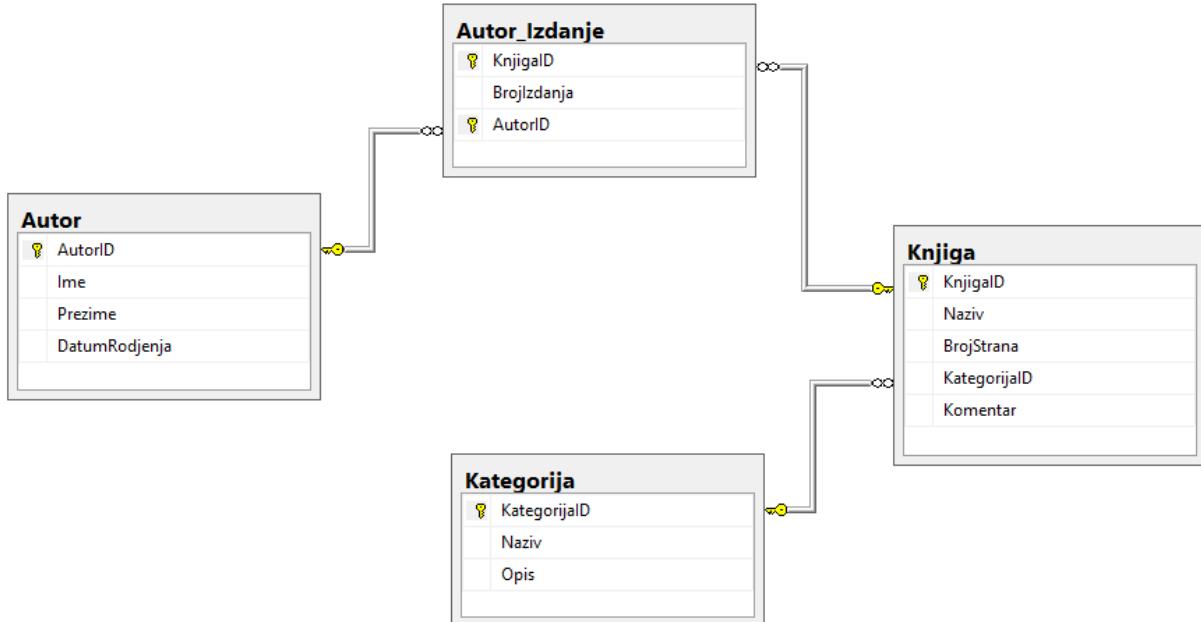
Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Онемогућено уношење погрешног временског интервала	5	0
Приказани подаци у оквиру са листом за изабрани комад	10	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б13: КОЛЕКЦИЈА КЊИГА

Књиге су саставни део нашег живота и као велики љубитељи користимо сваку прилику да нашу колекцију употребнимо новом књигом. Ради лакшег вођења евиденције креирали смо базу података у којој чувамо податке о њима.

Дијаграм базе је дат на слици:



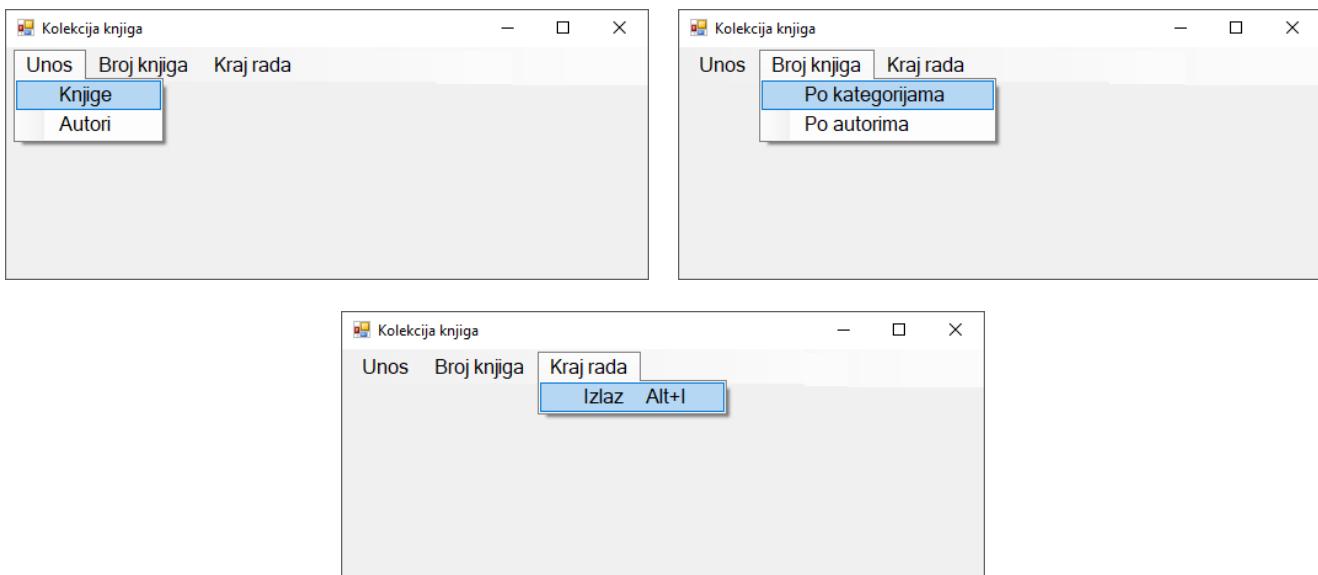
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

За сваку књигу потребно је да се води евиденција о називу издавача при чему један исти издавач може издати већи број књига.

Желимо да памтимо и број издања које је књига имала и цену сваког издања.

Волели би да у нашој бази чувамо и формате књиге односно њихову ознаку и кратак опис да би смо могли да боље организујемо нашу колекцију.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о књигама обавља се са преко обрасца чији је изглед дат на слици:

Упис књиге

Брисање књиге

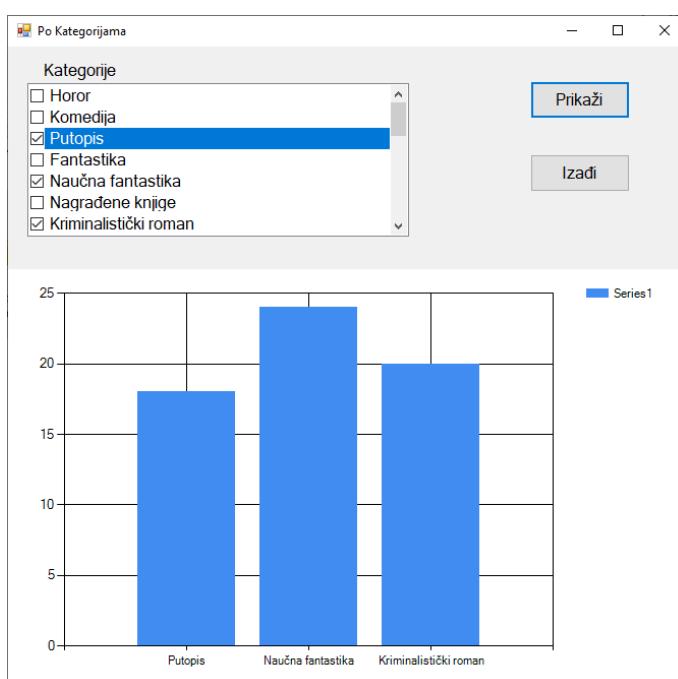
Број страница који се уноси мора бити цео број већи од нуле.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Po kategorijama**, графички приказује број књига изабраних категорија.

Образац треба да изгледа као на слици:



Називи категорија су сортирани у растући низ.

Корисник бира тачно три категорије књига.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б13

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б13				
Назив радног задатка	КОЛЕКЦИЈА КЊИГА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за назив	5	0
Постављена иницијална вредност за категорију	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 31)	ТАЧНО	НЕТАЧНО
Обезбеђена контрола избора акције за брисање/упис	4	0
Приказани подаци у одговарајућа поља ако унета шифра постоји	5	0
Обезбеђено уписивање података у базу по спецификацији задатка	10	0
Обезбеђено брисање података из базе по спецификацији задатка	8	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за категорију	5	0

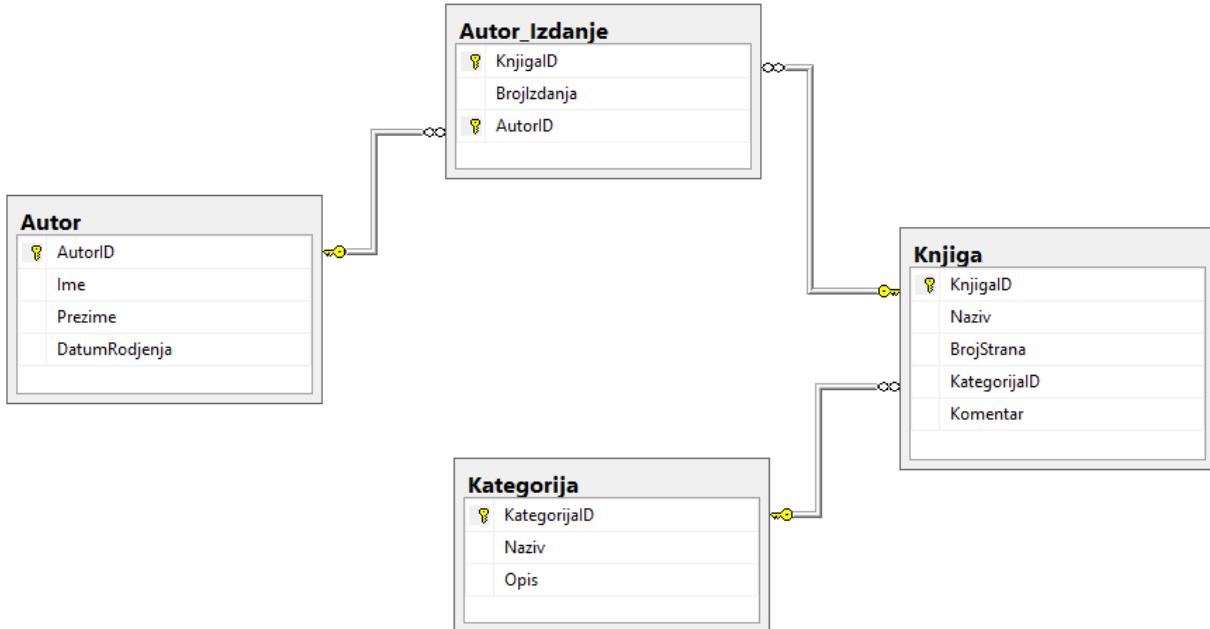
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Нацртан график са траженим подацима	15	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б14: КОЛЕКЦИЈА КЊИГА

Књиге су саставни део нашег живота и као велики љубитељи користимо сваку прилику да нашу колекцију употребнимо новом књигом. Ради лакшег вођења евиденције креирали смо базу података у којој чувамо податке о њима.

Дијаграм базе је дат на слици:



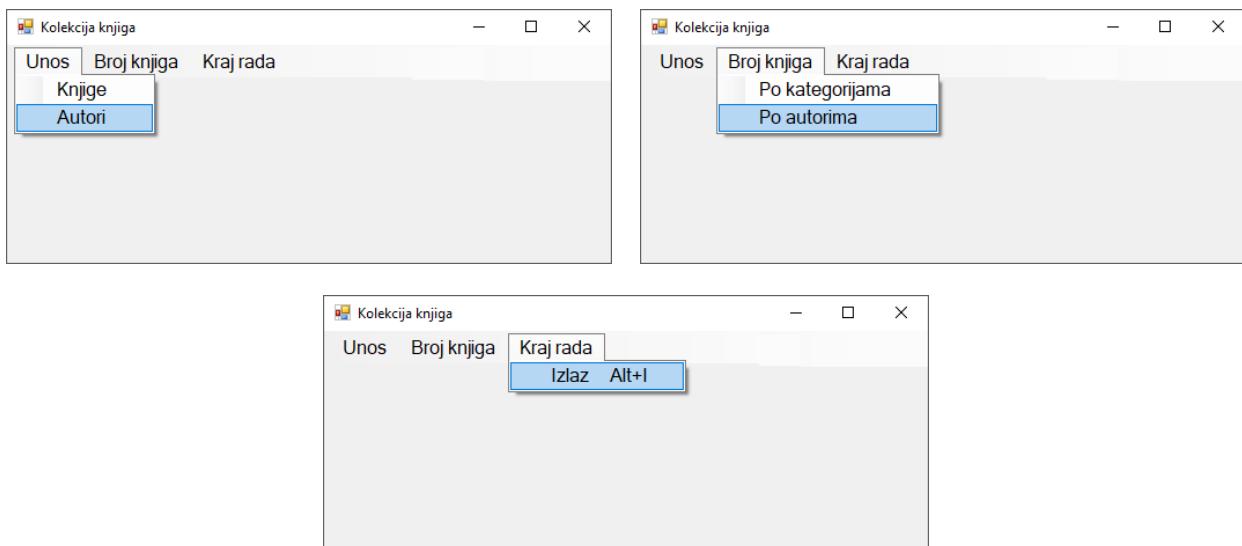
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

За сваку књигу потребно је да се води евиденција о називу издавача при чему један исти издавач може издати већи број књига.

Желимо да памтимо и број издања које је књига имала и цену сваког издања.

Волели би да у нашој бази чувамо и формате књиге односно њихову ознаку и кратак опис да би смо могли да боље организујемо нашу колекцију.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о ауторима обавља се преко обрасца чији је изглед дат на слици:

Упис аутора

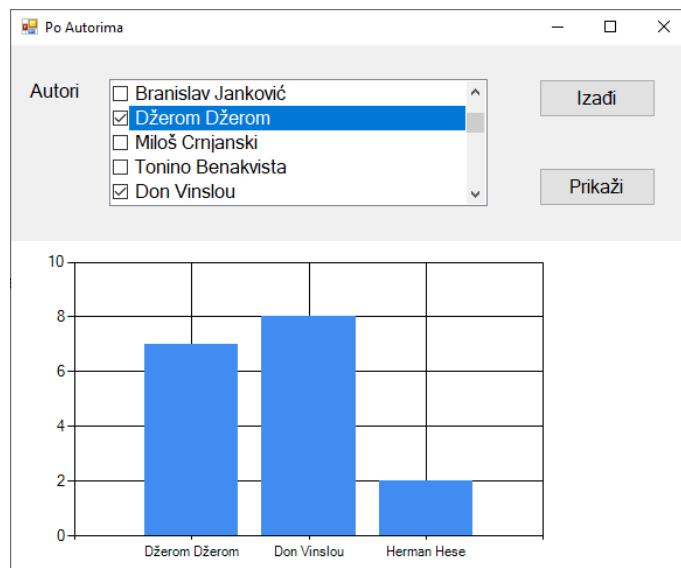
Брисање аутора

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Broj knjiga->Po autorima**, графички приказује број књига изабраних аутора.

Образац треба да изгледа као на слици:



Имена и презимена аутора сортирани су у растући низ.

Корисник бира тачно три аутора.

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираним базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б14

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б14				
Назив радног задатка	КОЛЕКЦИЈА КЊИГА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 7)	ПРАВИЛНО	НЕПРАВИЛНО
Креiran и оперативан мени	3	0
Креiran образац за унос података са траженим елементима	4	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 41)	ТАЧНО	НЕТАЧНО
Обезбеђена контрола избора акције за брисање/упис	4	0
Приказани подаци у одговарајућа поља ако унета шифра постоји	5	0
Онемогућено уношење датума рођења већег од данашњег	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	12	0
Обезбеђено брисање података из базе по спецификацији задатка	12	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ПРАВИЛНО	НЕПРАВИЛНО
Креiran образац са траженим елементима	4	0
Постављена иницијална вредност за аутора	5	0

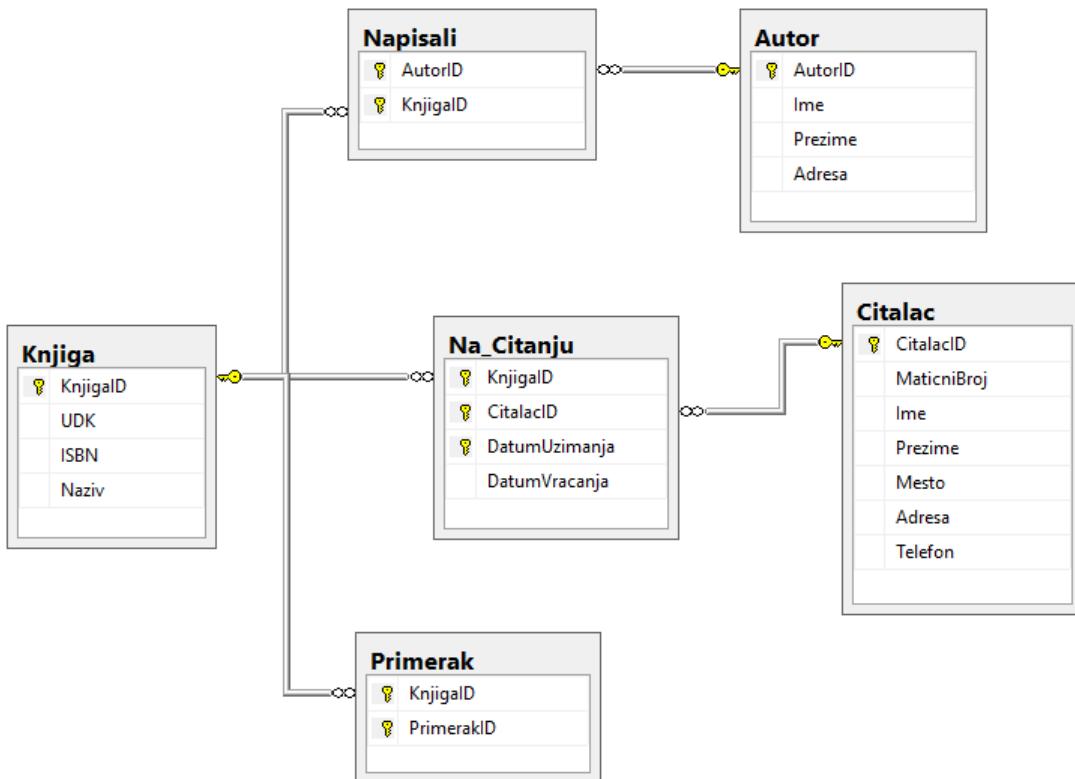
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Нацртан график са траженим подацима	15	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б15: ШКОЛСКА БИБЛИОТЕКА

Креирана је база података школске библиотеке.

Дијаграм базе података је дат на слици.



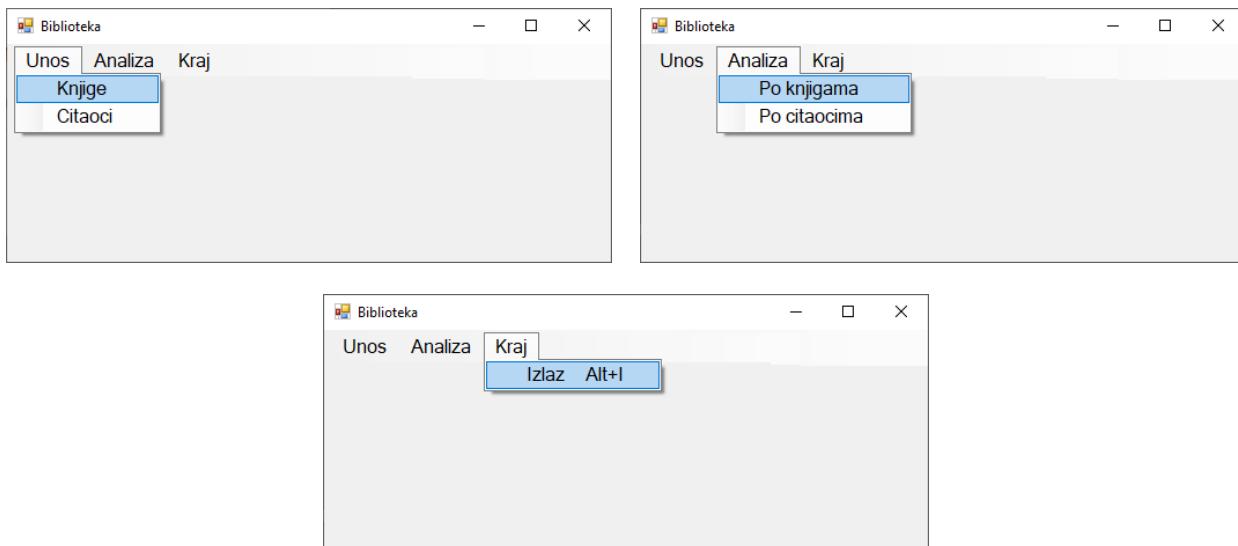
Користећи DDL наредбе, потребно је проширити базу тако да се задовоље следећи захтеви:

За аутора је потребно чувати и звање које има.

Књиге издају издавачи. Сваки издавач се карактерише називом и адресом.

Један издавач може да издаје више књига, а једна књига може да буде издата од стране више издавача. За свако издање памти се година издања.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о књигама обавља сепреко обрасца чији је изглед дат на слици:

Knjige

Šifra	15	Upiši
UDK	36333	Obrisi
ISBN	QYD-2398	Izmeni
Naziv	Braća Karamazovi	Izađi

Šifra	UDK	ISDN	Naziv
1	95099	GFG-7039	Majstor i Margarita
2	88060		Znakovi
3	97601	KVU-4347	Gradovi i himera
4		CIG-6322	Ararat
5	11222	MLC-2753	Pesme I

У оквиру са листом приказују се подаци о унетим књигама.

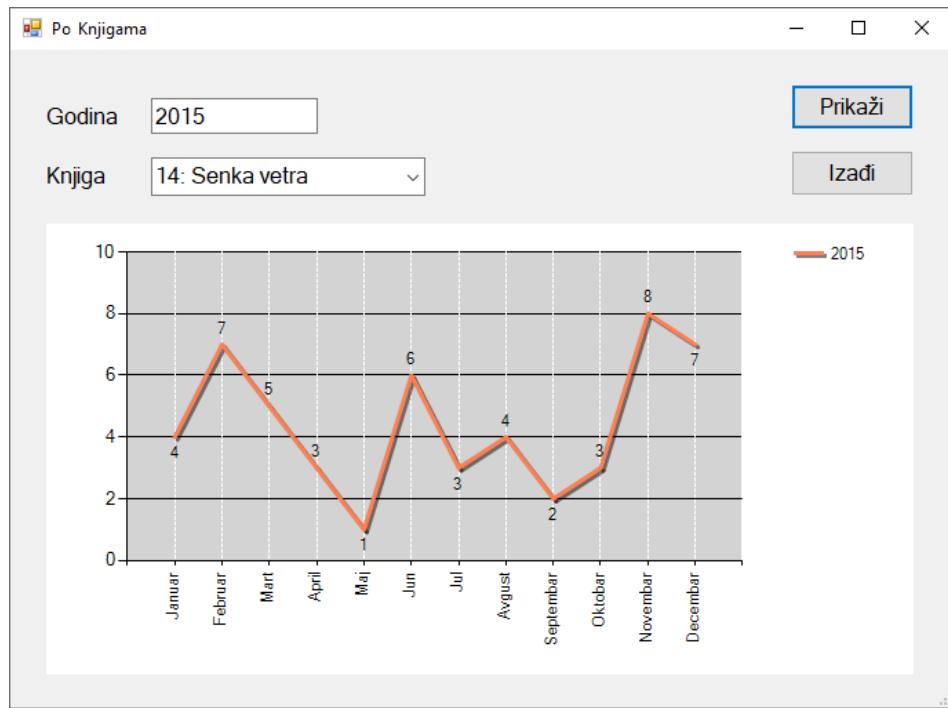
Ако унета шифра већ постоји у бази у одговарајућа поља приказују се подаци о тој књизи.

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Po knjigama**, графички приказује укупан број читалаца који су изнајмили одређену књигу, по месецима за одабрану календарску годину.

Образац треба да изгледа као на слици:



За књигу се приказују шифра и назив сортирани у растући низ.

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б15

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б15
Назив радног задатка	ШКОЛСКА БИБЛИОТЕКА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност у оквир са листом	8	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 37)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако унета шифра постоји	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	7	0
Обезбеђено брисање података из базе по спецификацији задатка	6	0
Обезбеђена измена података у бази по спецификацији задатка	7	0
Додата ставка у оквир са листомнакон уписа новог слога у базу	3	0
Обрисана ставка из оквира са листом након брисања слога из базе	3	0
Ажурирана ставка у оквиру са листомнакон ажурирања слога у бази	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 10)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за књигу	6	0

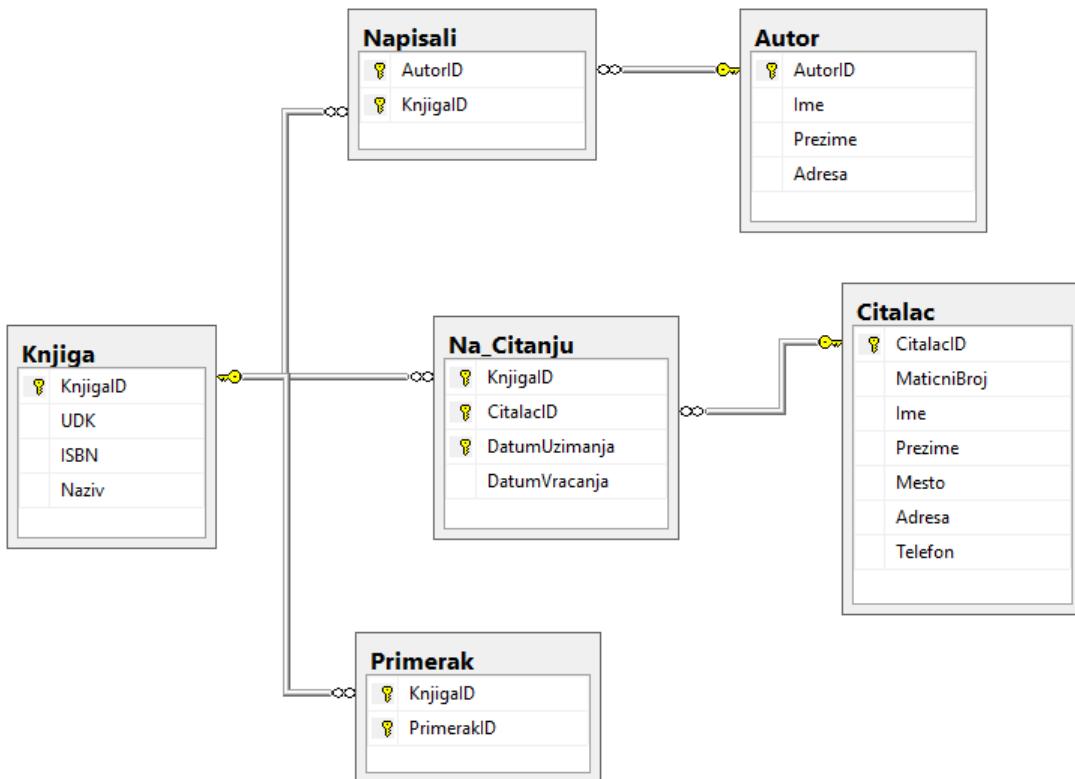
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Дефинисане вредности на х-оси графика	5	0
Нацртан график са траженим подацима	11	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б16: ШКОЛСКА БИБЛИОТЕКА

Креирана је база података школске библиотеке.

Дијаграм базе података је дат на слици.



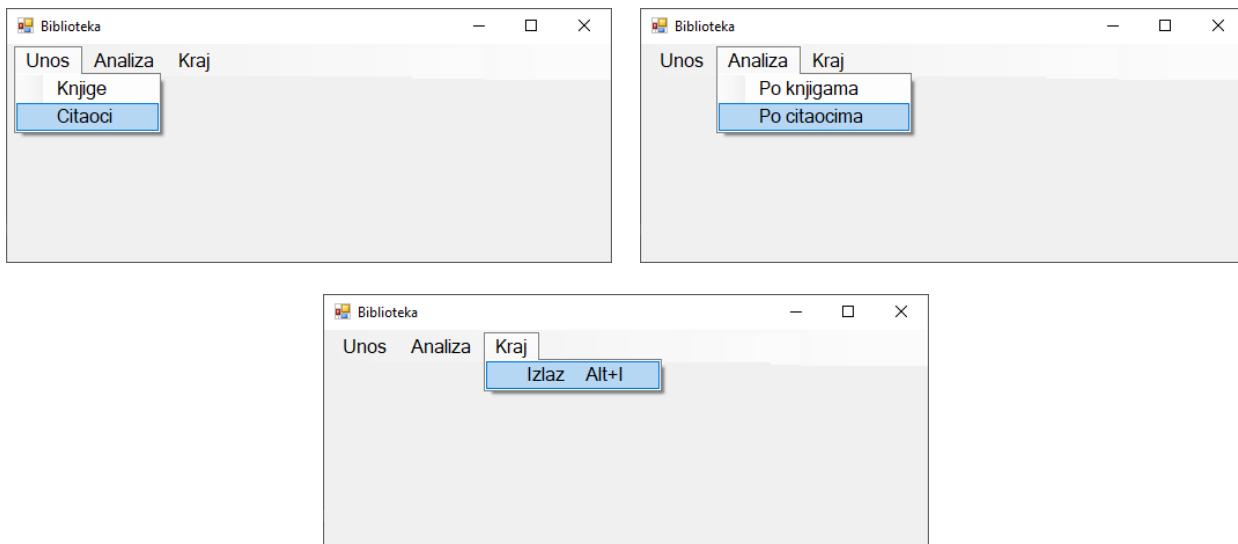
Користећи DDL наредбе, потребно је проширити базу је тако да се задовоље следећи захтеви:

За аутора је потребно чувати и звање које има.

Књиге издају издавачи. Сваки издавач се карактерише називом и адресом.

Један издавач може да издаје више књига, а једна књига може да буде издата од стране више издавача. За свако издање памти се година издања.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о читаоцима обавља се преко обрасца чији је изглед дат на слици:

Čitaoci

Šifra	<input type="text" value="7"/>	Upiši
Matični broj	<input type="text" value="1568172192445"/>	Obrisi
Ime	<input type="text" value="Đurđa"/>	Izmeni
Prezime	<input type="text" value="Avramović"/>	Izadi
Mesto	<input type="text" value="Sombor"/>	
Adresa	<input type="text"/>	
Telefon	<input type="text" value="355-761"/>	

Šifra	Matični broj	Ime	Prezime	Mesto	Adresa	Telefon
1	9327628972751	Jadranka	Babić	Smed...	Božidara ...	074-414-6739
2	2762131121271	Nataša	Apostolović	Kraguj...	Subotička ...	855-5084806
3	4727327683668	Staša	Dimitrijević	Sombor	Ravanička...	748-2378489
4	1441660569055	Vanja	Drenovac	Vlasot...	Vrežinska 5	757-913-9872
5	1126966884443	Majda	Erdeljan	Sombor	Branka Mil...	5181805879
6	5674693782602	Viktor	Avramović	Vršac	Subotička ...	1288165130
7	1568172192445	Đurđa	Avramović	Sombor		355-761

У оквиру са листом приказују се подаци о унетим читаоцима.

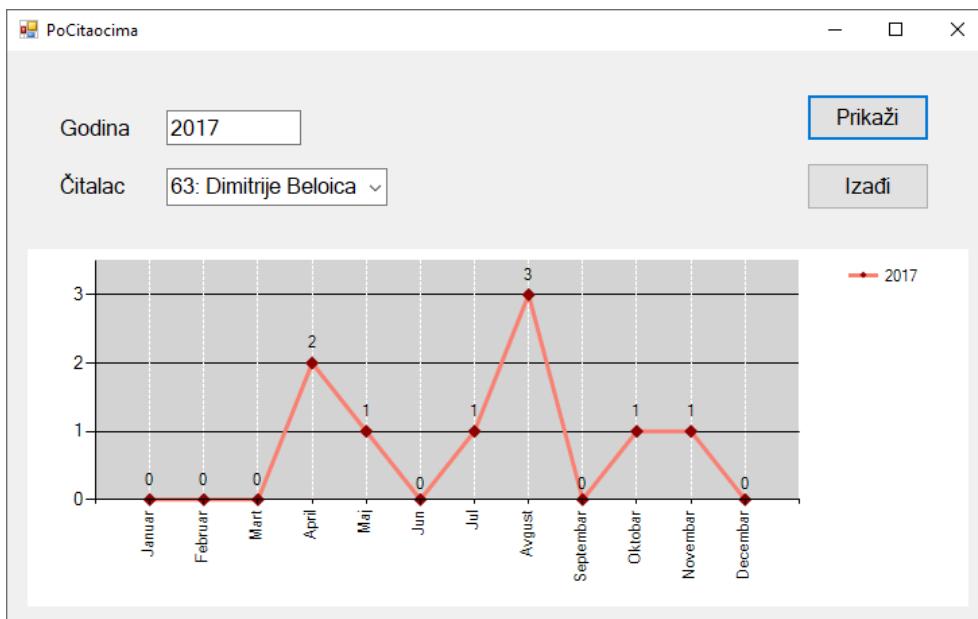
Ако унета шифра већ постоји у бази у одговарајућа поља приказују се подаци о том читаоцу.

Дугме **Izadi** затвара отворен образац и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Po citoocima**, графички приказује укупан број књига који је изнајмио одређени читаоц, по месецима за одабрану календарску годину.

Образац треба да изгледа као на слици:



За читаоца се приказује шифра и име и презиме сортирани у растући низ.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б16

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б16				
Назив радног задатка	ШКОЛСКА БИБЛИОТЕКА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност у оквир са листом	8	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 37)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако унета шифра постоји	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	7	0
Обезбеђено брисање података из базе по спецификацији задатка	6	0
Обезбеђена измена података у бази по спецификацији задатка	7	0
Додата ставка у оквир са листом	3	0
Обрисана ставка из оквира са листом	3	0
Ажурирана ставка у оквиру са листом	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 10)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за читаоца	6	0

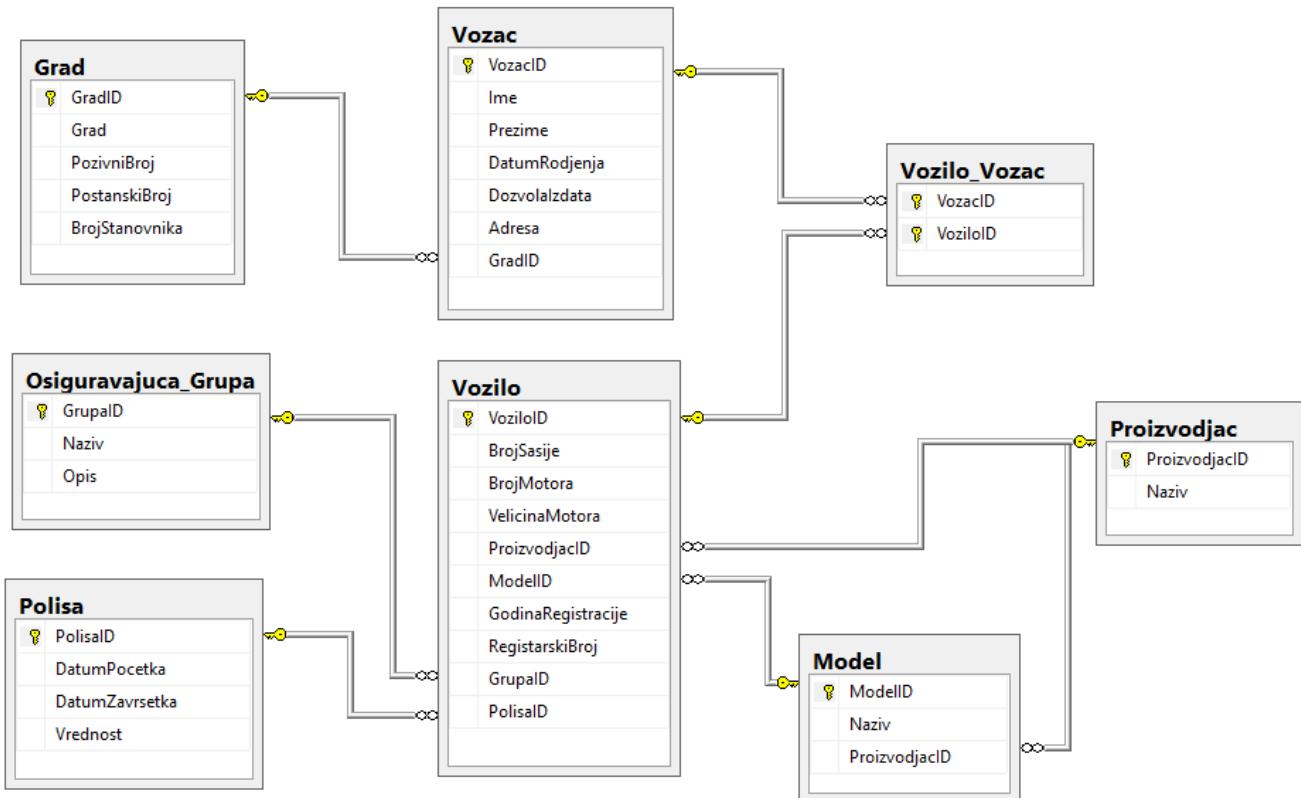
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Дефинисане вредности на x-оси графика	5	0
Нацртан график са траженим подацима	11	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б17: ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА

За потребе осигуравајућег друштва креирана је база података.

Дијаграм базе података је дат на слици:

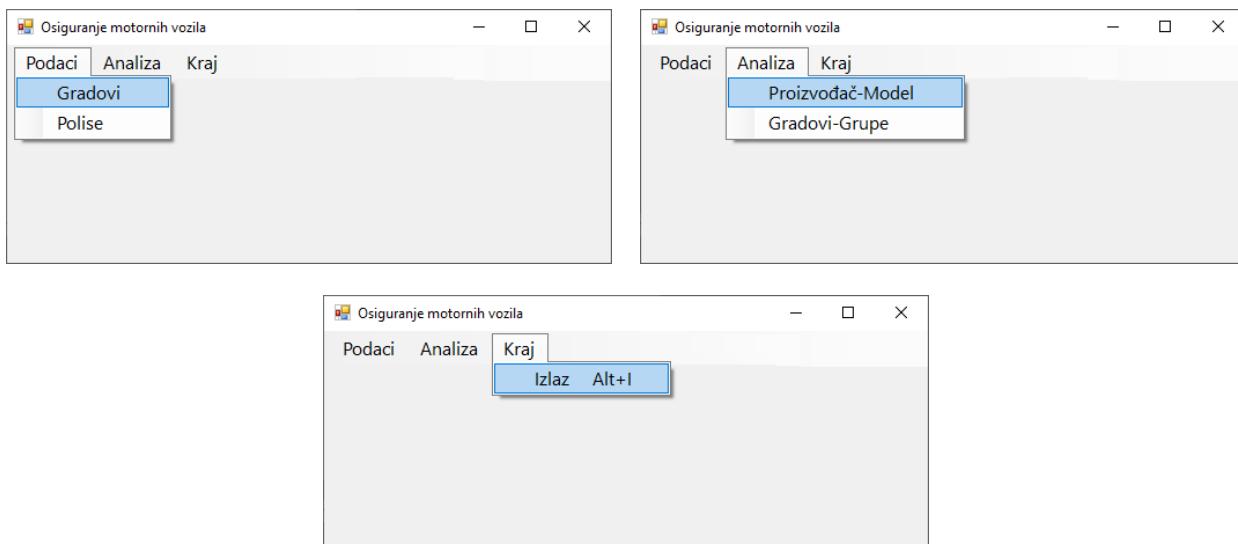


Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

У табели возило желимо да чувамо и тренутну вредност возила.

Вредност полисе одређује се и на основу листе прекршаја које је возач направио, а која се добија од полиције. Листа прекршаја садржи назив и опис. Један возач може да направи ниједан или више прекршаја, а више возача може да направи један исти прекршај.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о градовима обавља се преко обрасца чији је изглед дат на слици:

Ако унета шифра постоји приказују се подаци о граду.

Када се унесе поштански број, кориснику се аутоматски нуди позивни број који одговара том поштанском броју (нпр. 18000 и 018, 21123 и 021).

Број становника који се уноси мора бити цео број већи од нуле.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Proizvodjac-model**, приказује листу шифара, имена и презимена власника возила, регистарски број возила, шифру полисе и датум почетка и датум завршетка важења полисе, за сва возила која припадају одговарајућем произвођачу и моделу возила, и за одабран статус полисе.

Образац треба да изгледа као на слици:

	Šifra vozača	Ime	Prezime	Registracija	PolisalD	Važi od
>	605	Milan	Filipović	GV-678-VN	6	02.10.2002
	512	Uroš	Avramović	IJ-389-ZX	278	28.01.2006
	200	Mihajlo	Borovac	EE-350-TN	560	26.10.2006
	447	Jadranka	Hajdeman	ES-157-PG	563	29.09.2006
	2	Olivera	Andrejić	KP-849-NP	562	30.08.2008
	478	Dunja	Dimitrijević	AH-878-IY	4	15.07.2010
	163	Dunja	Danilović	DI-506-LJ	559	06.11.2016

За произвођача се приказује прва ставка **Sve**, а затим шифре и називи производаца сортирани у растући низ.

За модел се приказује прва ставка **Sve**, а затим шифре и називи модела сортирани у растући низ.

Подаци се приказују у контроли за табеларни приказ података.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б17

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б17				
Назив радног задатка	ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за шифру града	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 37)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља по избору шифре	4	0
Креиран позивни број на основу поштанског броја	3	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Обезбеђено брисање података из базе по спецификацији задатка	8	0
Обезбеђено ажурирање података у бази по спецификацији задатка	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 14)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за произвођача	5	0
Постављена иницијална вредност за модел	5	0

5. Креирање дела апликације за упит из базе

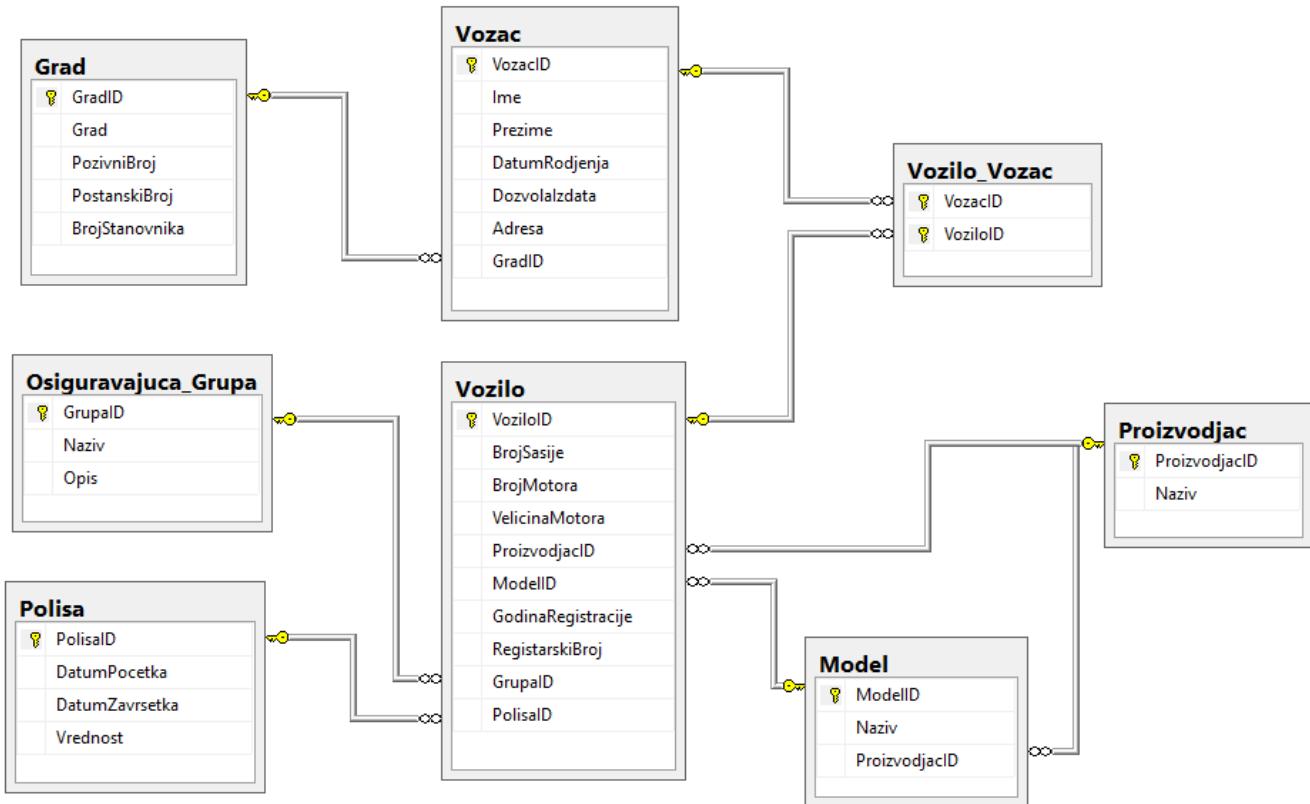
ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Приказани подаци у контроли за табеларни приказ података	15	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б18
НАЗИВ РАДНОГ ЗАДАТКА: ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА

ЕИТ-Б18: ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА

За потребе осигуравајућег друштва креирана је база података.

Дијаграм базе података је дат на слици:

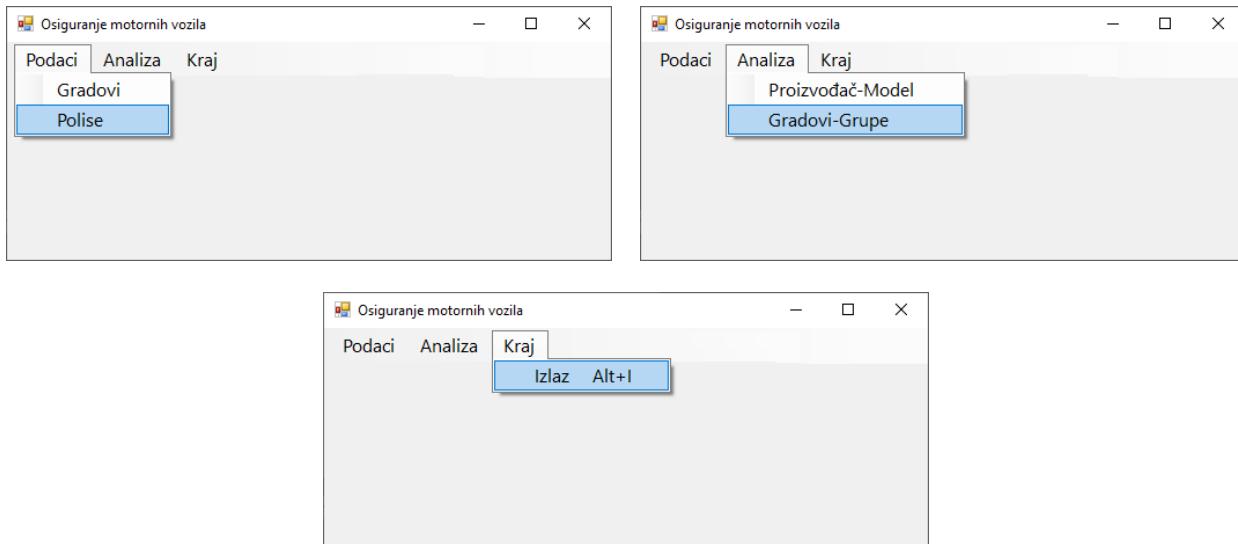


Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

У табели возило желимо да чувамо и тренутну вредност возила.

Вредност полисе одређује се и на основу листе прекршаја које је возач направио, а која се добија од полиције. Листа прекршаја садржи назив и опис. Један возач може да направи ниједан или више прекршаја, а више возача може да направи један исти прекршај.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о полисама обавља се преко обрасца чији је изглед дат на слици:

Вредност полисе која се уноси мора бити ненегативан цео број.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Gradovi-grupe**, приказује листу шифара, имена и презимена власника возила, регистарски број возила, шифру полисе и датум почетка и датум завршетка важења полисе, за сва возила која су из одговарајућег града, и за одабран тип и статус полисе.

Образац треба да изгледа као на слици:

	Šifra vozača	Ime	Prezime	Registracija	PolisID	Važi od	Važi do
▶	273	Brankica	Antić	OK-758-OE	188	04.05.2013	26/11/2013
	565	Dunja	Avramović	CC-428-YB	147	24.05.2013	01/01/2014
	72	Lidija	Bajić	LV-829-DT	195	20.02.2012	19/11/2012
	77	Aleksandra	Apostolović	UQ-867-AX	171	06.03.2008	07/09/2008
	273	Brankica	Antić	GD-572-YR	190	08.12.2003	18/06/2004
	849	Ibrahim	Erdeljan	CO-716-JR	187	10.11.2017	19/09/2018
	520	Milena	Beloica	QZ-324-OS	150	07.08.2003	04/02/2004

За групу осигурања приказује се прва ставка **Sve**, а затим шифре и називи сортирани у растући низ.

За град се приказује прва ставка **Sve**, а затим шифре и називи градова сортирани у растући низ.

Подаци се приказују у контроли за табеларни приказ података.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б18

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б18				
Назив радног задатка	ОСИГУРАЊЕ МОТОРНИХ ВОЗИЛА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за шифру полисе	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 37)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља по избору шифре полисе	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Обезбеђено брисање података из базе по спецификацији задатка	8	0
Обезбеђено ажурирање података у бази по спецификацији задатка	9	0
Обрађена упозорења за опсег датума	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 14)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за групу осигурања	5	0
Постављена иницијална вредност за град	5	0

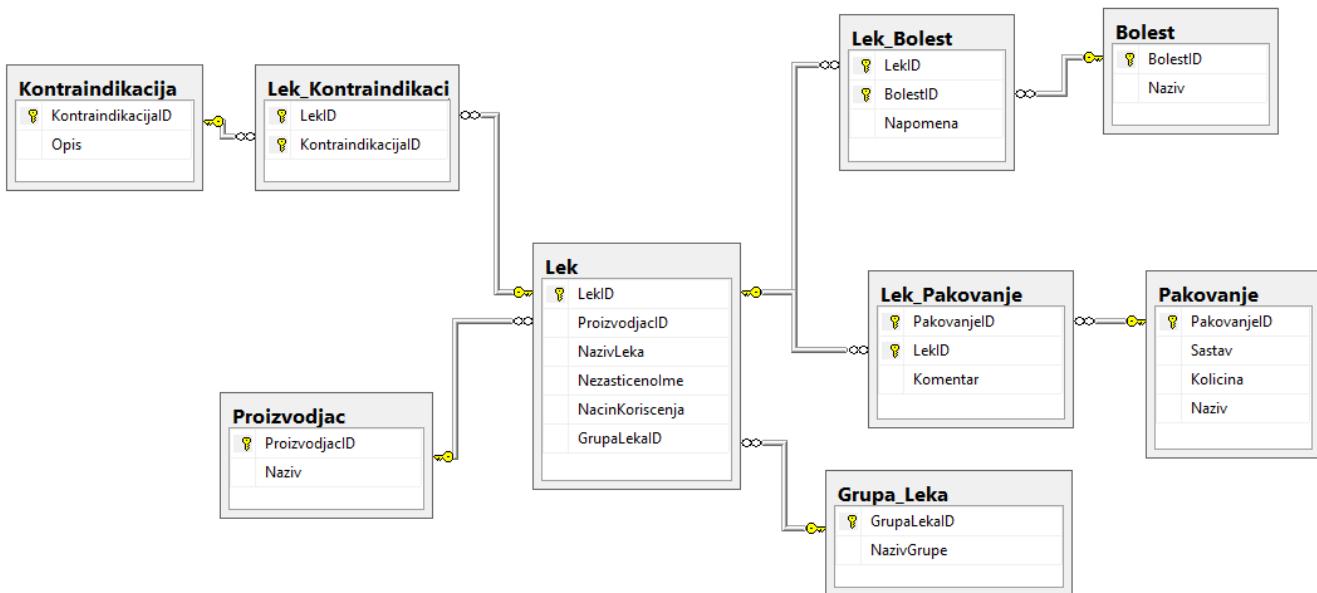
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Приказани подаци у контроли за табеларни приказ података	15	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б19: ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ

Креирана је база за евиденцију лекова у апотеци.

Дијаграм базе података је дат на слици.



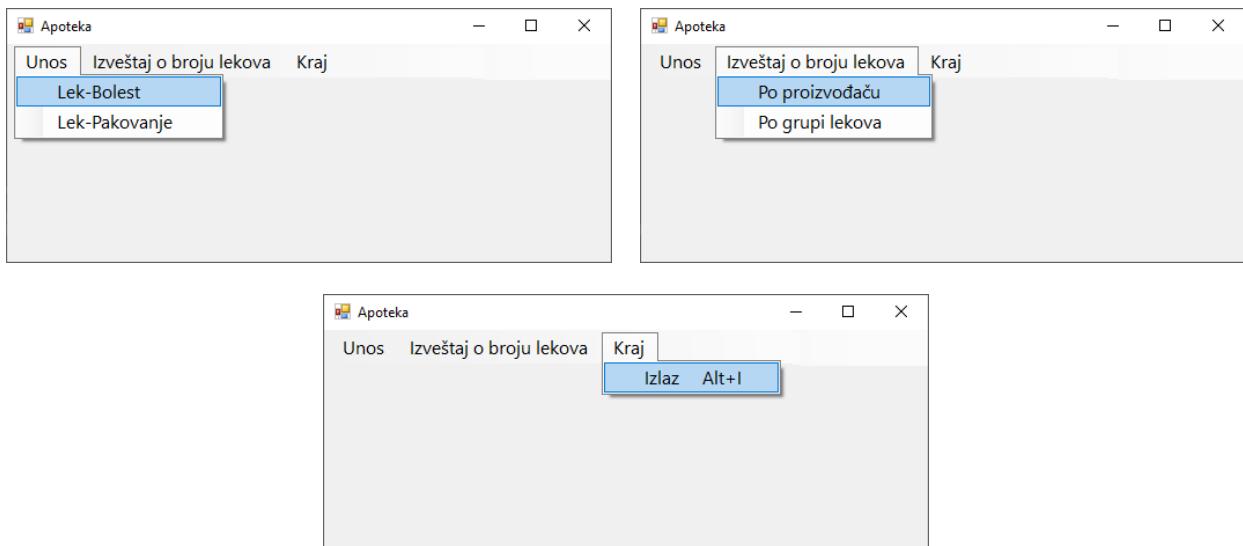
Користећи DDL наредбе проширибазу тако да се задовоље следећи захтеви:

У табели паковање желимо дачувамо податке и о облику паковања.

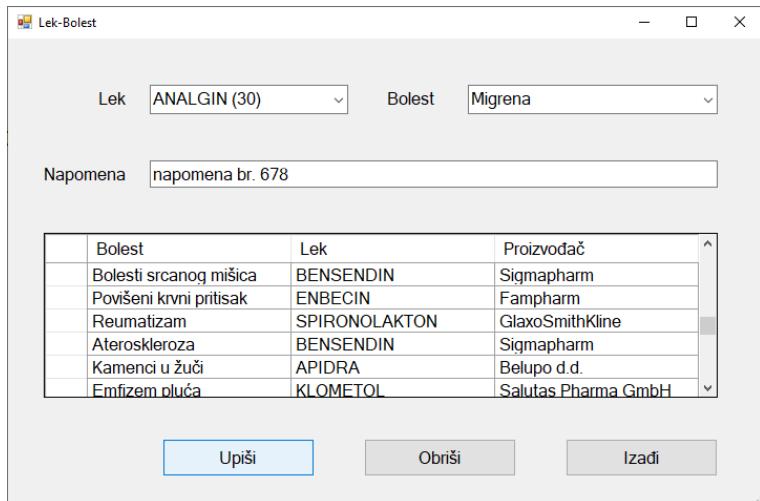
Сваки лек може да има више од једног нежељеног ефекта; један ефекат може да буде карактеристика више лекова.

Нежељени ефекат осим своје шифре има и свој опис.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о томе који се лекови додељују којим болестима, обавља се преко обрасца чији је изглед дат на слици:



За лек се приказује шифра лека и назив лека, сортирани у растући низ.

Називи болести треба да буду сортирани у растућем абецедном низу.

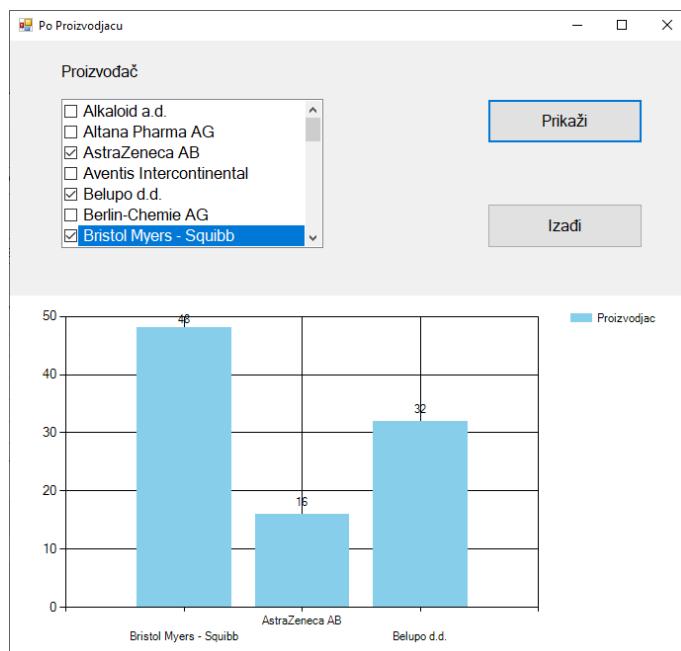
У контроли за табеларни приказ података приказују се подаци који постоје у бази.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Обезбедите да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Po proizvodjacu**, графички приказује укупан број различитих болести за које изабрани производијач производи лекове.

Образац треба да изгледа као на слици:



Називи свих производијача лекова сортирани су у растућем абецедном низу.

Корисник бира тачно три производијача.

Дугме **Izadjii** затвара отворен образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б19

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б19				
Назив радног задатка	ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 25)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за лек	5	0
Постављена иницијална вредност за болест	5	0
Постављена иницијална вредност у контроли за табеларни приказ података	8	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 28)	ТАЧНО	НЕТАЧНО
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Додата ставка у контроли за табеларни приказ података након уписа слога у базу	4	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Обрисана ставка из контроле за табеларни приказ података након брисања слога из базе	4	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за произвођача	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Нацртан график са траженим подацима	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

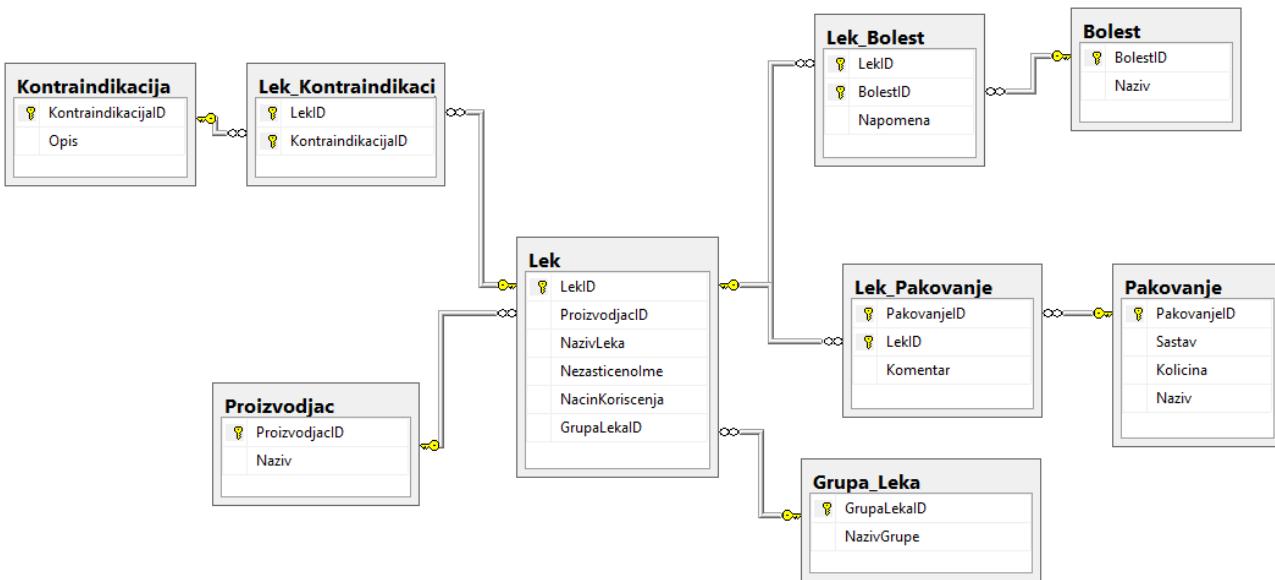
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б20

НАЗИВ РАДНОГ ЗАДАТКА: ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ

ЕИТ-Б20: ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ

Креирана је база за евиденцију лекова у апотеци.

Дијаграм базе података је дат на слици.



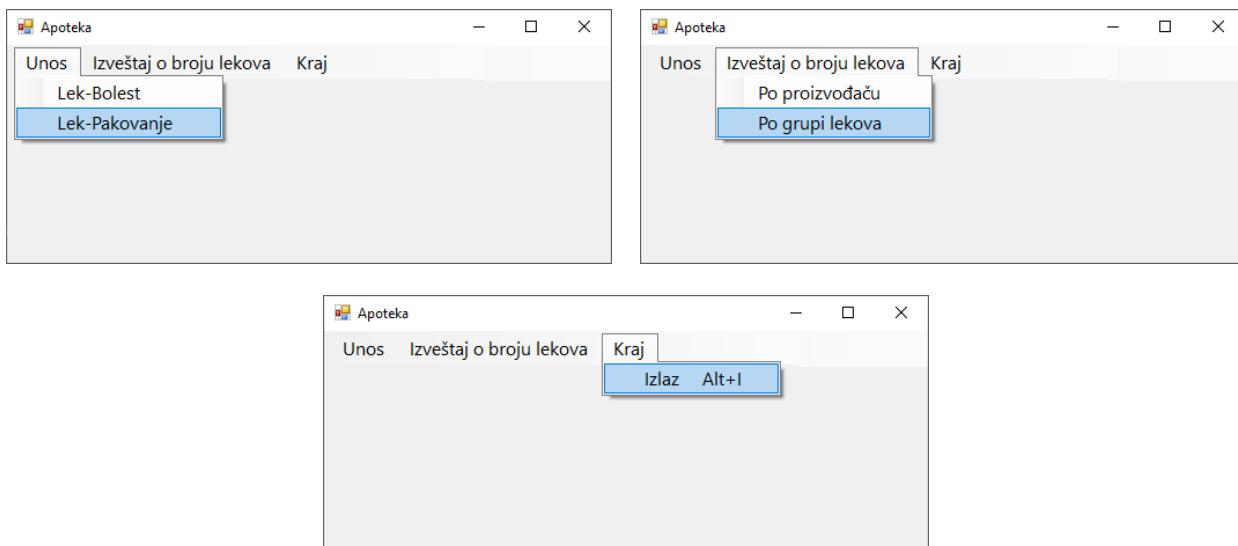
Користећи DDL наредбе проширибазу тако да се задовоље следећи захтеви:

У табели паковање желимо дачувамо податке и о облику паковања.

Сваки лек може да има више од једног нежељеног ефекта; један ефекат може да буде карактеристика више лекова.

Нежељени ефекат осим своје шифре има и свој опис.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим слицима:



Уношење података о томе који се лекови додељују којим паковањима обавља се преко обрасца чији је изглед дат на слици:

Lek-pakovanje

Lek: HEMOMYCIN (19) Pakovanje: kapsula

Komentar: komentar br. 134

	Naziv leka	Proizvođač	Pakovanje	Količina
▶	HUMULIN R	Pfizer Enterprises	kapi	5
	HUMULIN R	Pfizer Enterprises	obložena tableta	22
	DIFUTRAT	Remevita d.o.o.	rastvor za kožu	2
	DIFUTRAT	Remevita d.o.o.	injekcija,karpula	20
	BRUFEN	Altana Pharma AG	sirup	23
	BRUFEN	Altana Pharma AG	kapsula	1

Upiši Obriši Izadi

За лек се приказују шифра лека и назив лека, сортирани у растући низ.

За паковање се приказују шифра паковања, назив и количина.

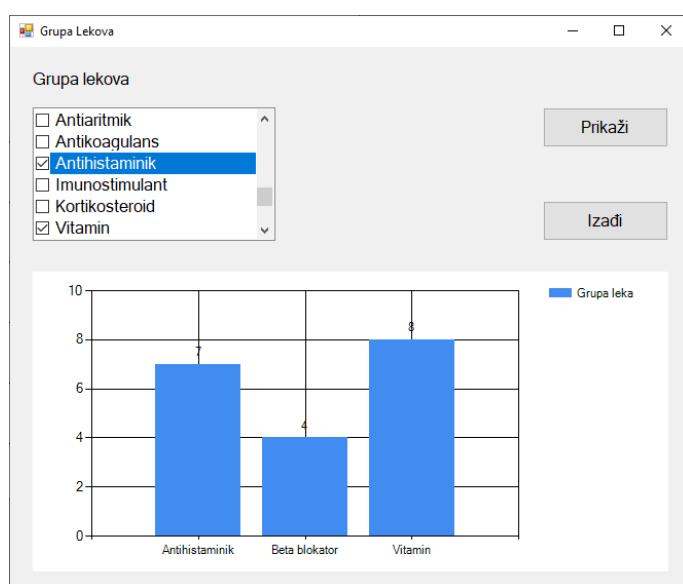
У контроли за табеларни приказ података приказују се подаци који постоје у бази.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Po grupi lekova**, графички приказује укупан број различитих лекова који припадају изабраним групама лекова.

Образац треба да изгледа као на слици:



Називи свих група лекова сортирани су у растући абецедни низ.

Корисник бира тачно три групе лекова.

Дугме **Izadjii** затвара отворен образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б20

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б20				
Назив радног задатка	ЕВИДЕНЦИЈА ЛЕКОВА У АПОТЕЦИ				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 25)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за лек	5	0
Постављена иницијална вредност за паковање	5	0
Постављена иницијална вредност у контроли за табеларни приказ података	8	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 28)	ТАЧНО	НЕТАЧНО
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Добавата ставка у контроли за табеларни приказ података након уписа новог слога у базу	4	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Обрисана ставка из контроле за табеларни приказ података након брисања слога из базе	4	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за произвођача	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Нацртан график са траженим подацима	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

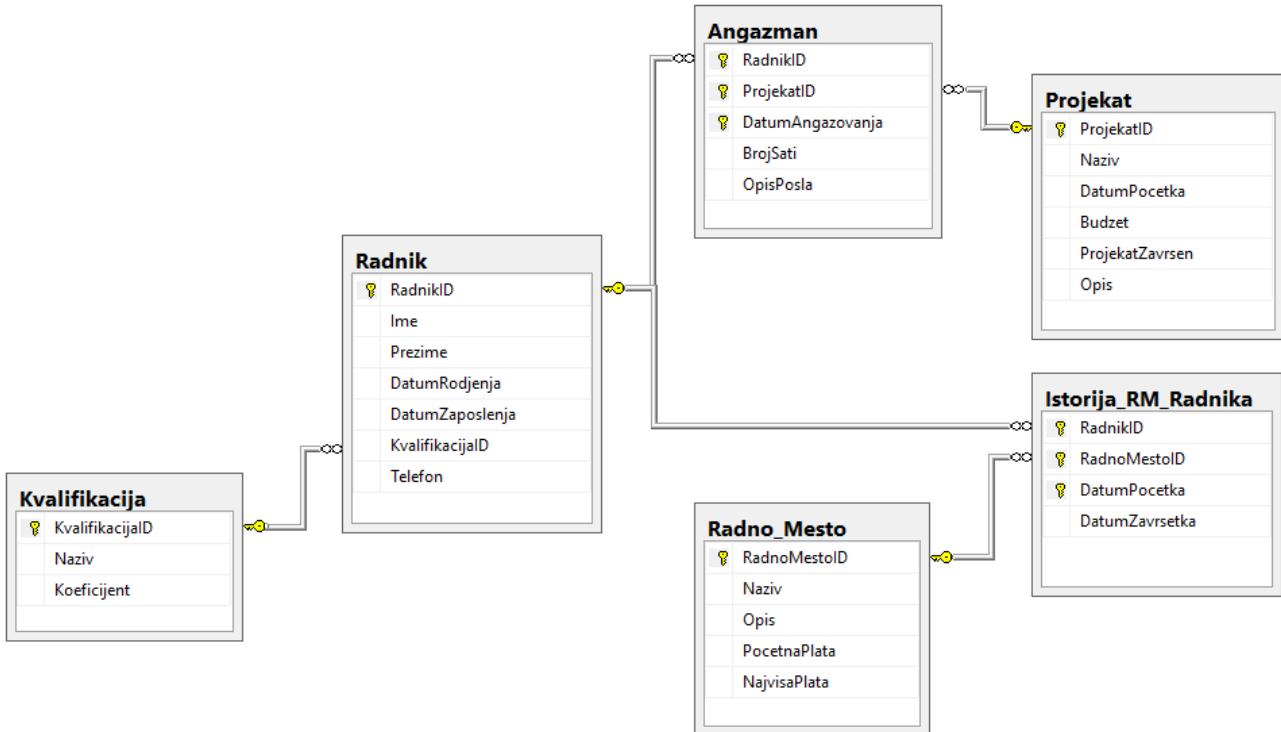
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б21

НАЗИВ РАДНОГ ЗАДАТКА: ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА

ЕИТ-Б21: ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА

Креирана је база података у коју ће се уносити подаци везани за пројекте на којим раде запослени у компанији.

Дијаграм базе је дат на слици:



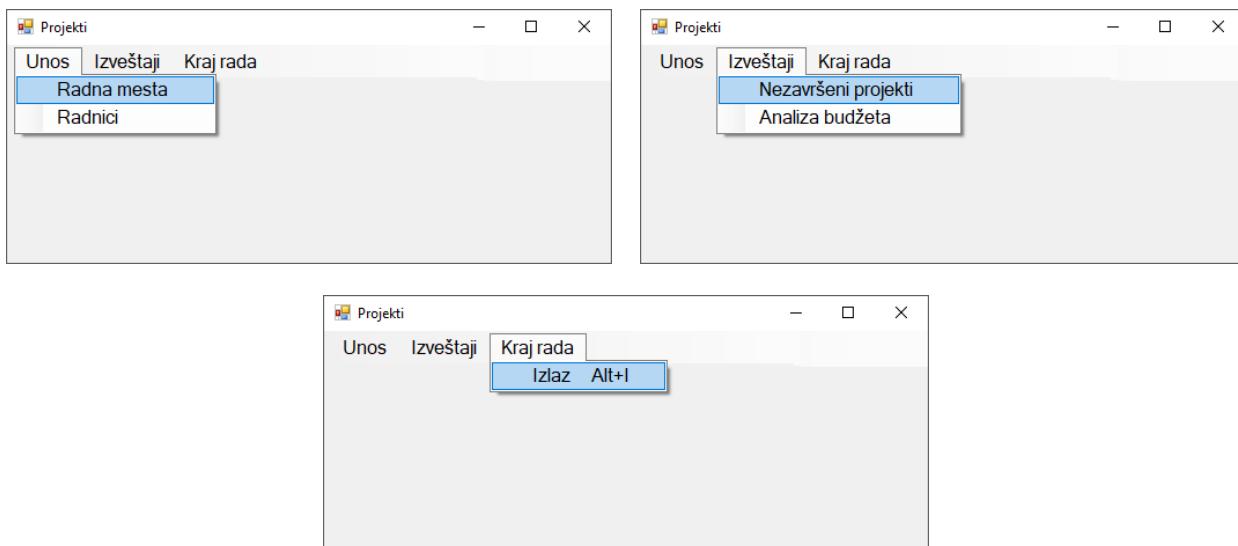
Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

Радници су смештени у канцеларије, које се карактеришу бројем, величином, телефоном и шифром сектора којем припадају. Један радник може да буде смештен само у једној канцеларији, а у једној канцеларији може да се налази више радника.

Сектор се карактерише називом и описом делатности. Један радник ради у једном сектору, а један сектор има више радника.

Сваки сектор има шефа сектора који је такође радник.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о радним местима обавља се путем образца чији је изглед дат на сликама:

Упис радног места

Брисање радног места

Измена радног места

Обезбедити да се у поља за почетну и највишу плату не уносе негативне вредности.

Дугме **Izadjii** затвара отворен образац и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Nezavrseni projekti**, приказује листу шифара и назива пројекта који нису завршени сортирани по шифри пројекта у растући низ.

Образац треба да изгледа као на слици:

Sifra	Naziv	Godine	Meseci	Dana	Datum pocet
2	Jačanje lokalne samo...	11	4	1	27.08.2009
3	Mehanizam odgovorn...	11	7	11	17.05.2009
4	ICT psp	11	3	29	29.08.2009
7	Jačanje zaštite potrošača	4	6	12	16.06.2016
11	Program eu exchange	4	9	6	12.06.2011
14	Eu - horizon 2020	8	10	18	10.02.2012
15	Otvoreni zagrljaj	8	11	21	07.01.2012
16	Digitalizacija	10	6	29	29.05.2010
19	CBC Srbija - Bosna i ...	7	4	29	29.07.2013
21	Druga šansa	10	6	17	11.06.2010
22	CBC Srbija-Crna Gora	11	5	5	23.07.2009
25	ICIP	10	5	20	08.07.2010
26	Eu progres	10	10	14	14.02.2010
27	Projekat podrške inov...	11	3	16	12.09.2009
36	Fabrika za flaširanje n...	10	1	6	22.11.2010

Sifra	Radnik
2	Darija Danil...
17	Branka Filip...
18	Andrej Dimit...
44	Ljiljana Erde...
45	Petra Bebić
140	Lidija Aleksić
142	Tadija Boga...
167	Ana Bogavac
168	Ana Bogavac
246	Vanja Erceg

Резултате упита приказати у контроли за приказ података у облику листе. За сваки пројекат приказује се време протекло од почетка пројекта до данашњег дана.

Када корисник изабере неки од пројекта, уконтроли за табеларни приказ података приказује се листа шифара и имена и презимена свих радника који су ангажовани на том пројекту.

Дугме **Izadjii** затвара отворен образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б21

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б21
Назив радног задатка	ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за шифру	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Обезбеђена контрола избора акције за упис/брисање/измену	4	0
Приказане вредности за одабрану шифру уколико слог постоји	5	0
Обезбеђено уписивање података у базу по спецификацији задатка	7	0
Обезбеђено брисање података из базе по спецификацији задатка	5	0
Обезбеђена измена података у бази по спецификацији задатка	7	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност у контроли за приказ података у облику листе (шифра, назив и трајање пројекта)	10	0
Трајање пројекта приказано у листи у облику година, месец и дан	6	0

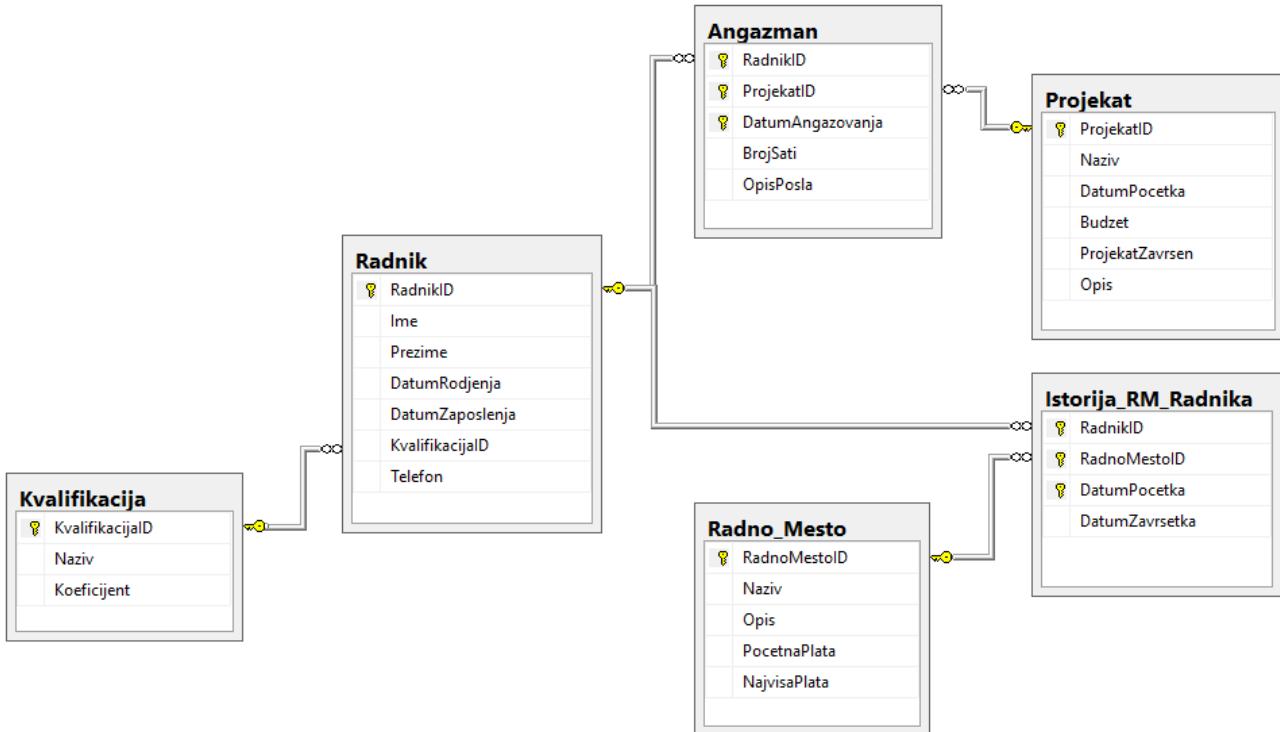
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Приказана листа радника за изабран пројекат	10	0
Обрађене грешке	2	0

ЕИТ-Б22: ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА

Креирана је база података у коју ће се уносити подаци везани за пројекте на којим раде запослени у компанији.

Дијаграм базе је дат на слици:



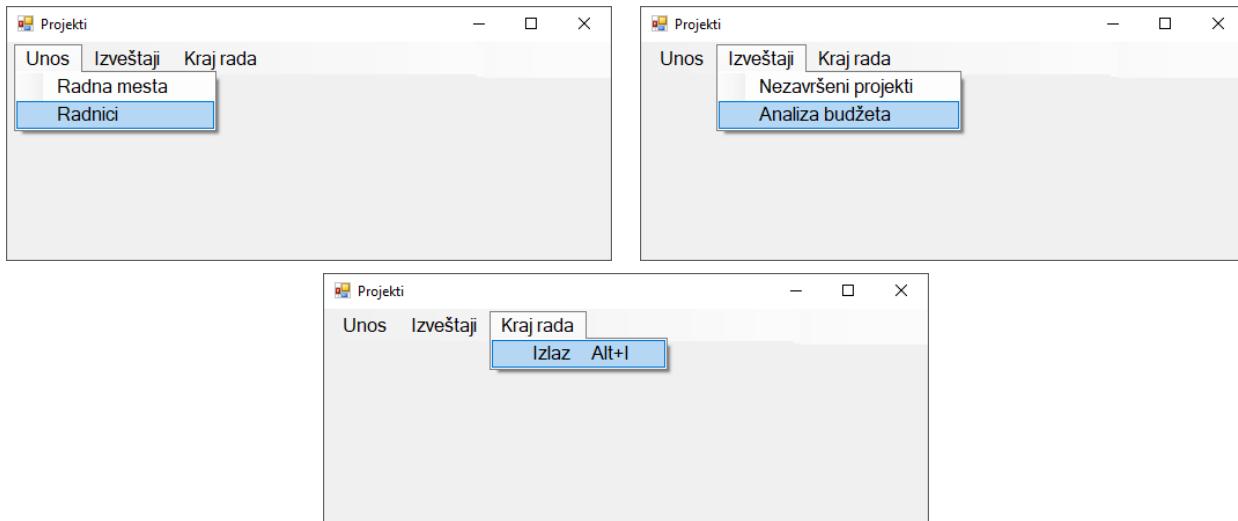
Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

Радници су смештени у канцеларије, које се карактеришу бројем, величином, телефоном и шифром сектора којем припадају. Један радник може да буде смештен само у једној канцеларији, а у једној канцеларији може да се налази више радника.

Сектор се карактерише називом и описом делатности. Један радник ради у једном сектору, а један сектор има више радника.

Сваки сектор има шефа сектора који је такође радник.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о радницима обавља се преко образца чији је изглед дат на сликама:

Упис радника

Брисање радника

Измена радника

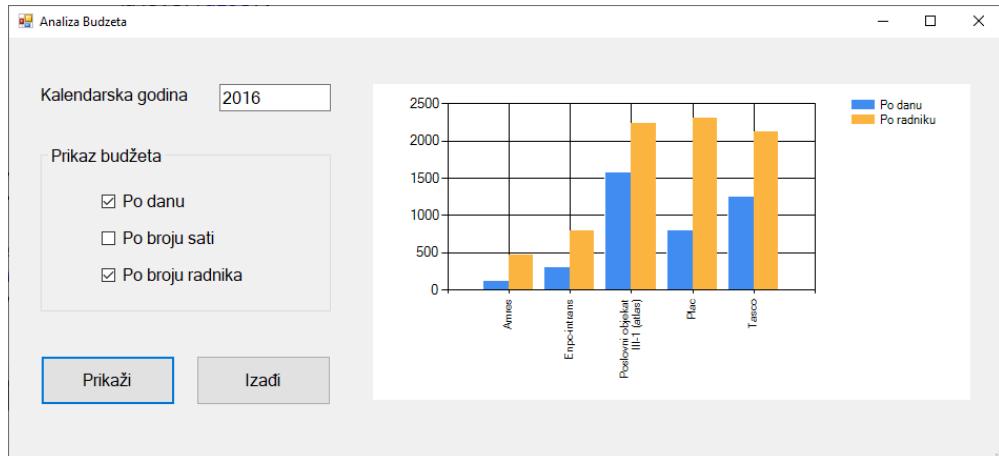
Обезбедити да корисник не може да унесе неважеће датуме – 31. јун и сл.

Дугме **Izadj** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Analiza budžeta**, графички приказује анализу буџета завршених пројеката који су започели са реализацијом у одређеној години.

Образац треба да изгледа као на слици:



Дугме **Izadj** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б22

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б22
Назив радног задатка	ЕВИДЕНЦИЈА РАДНИКА НА ПРОЈЕКТИМА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 15)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за дан и месец у контролу комбинованог оквира за текст са листом	3	0
Постављена иницијална вредност за квалификацију	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 34)	ТАЧНО	НЕТАЧНО
Обезбеђена контрола избора акције за упис/брисање/измену	4	0
Приказане вредности за одабрану шифру уколико слог постоји	5	0
Обезбеђено уписивање података у базу по спецификацији задатка	7	0
Обезбеђено брисање података из базе по спецификацији задатка	5	0
Обезбеђена измена података у бази по спецификацији задатка	7	0
Онемогућен унос неважећег датума	2	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 4)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 23)	ТАЧНО	НЕТАЧНО
Дефинисане тражене вредности на x-оси графика	5	0
Нацртан график са траженим подацима	16	0
Обрађене грешке	2	0

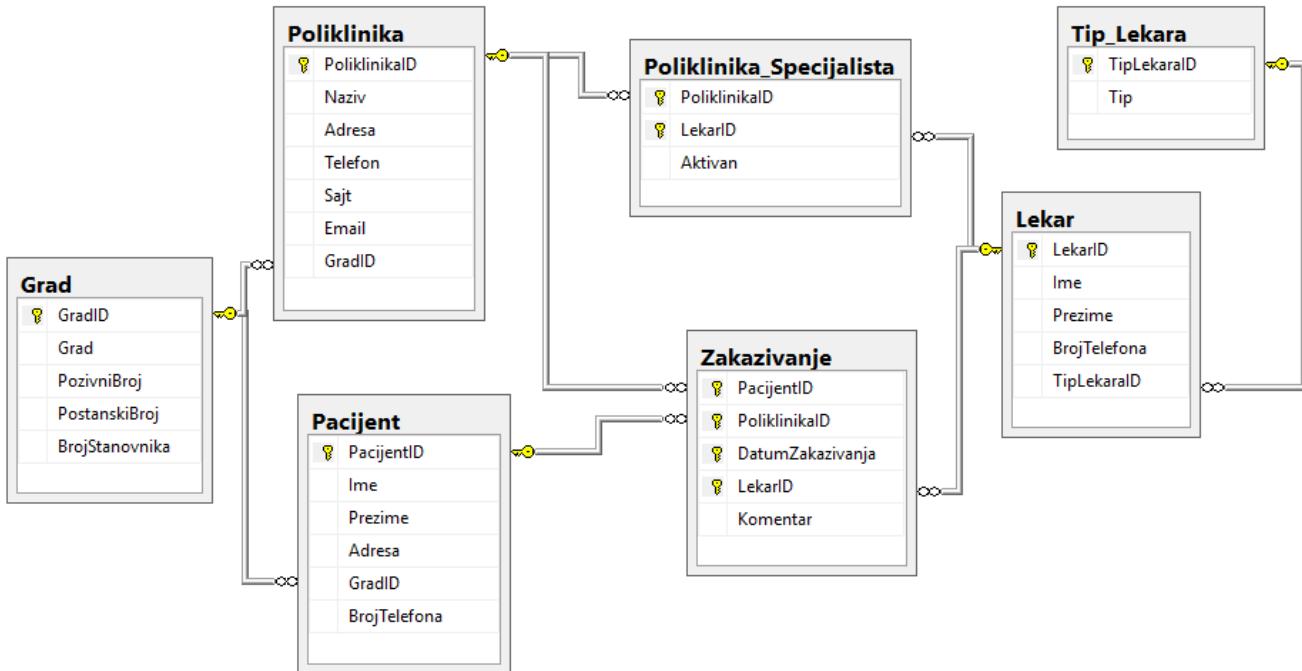
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б23

НАЗИВ РАДНОГ ЗАДАТКА: ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ

ЕИТ-Б23: ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ

Креирана је база података у којој се води евиденција о пацијентима који обављају прегледе на поликлиникама и њиховим лекарима.

Дијаграм базе је дат на слици:

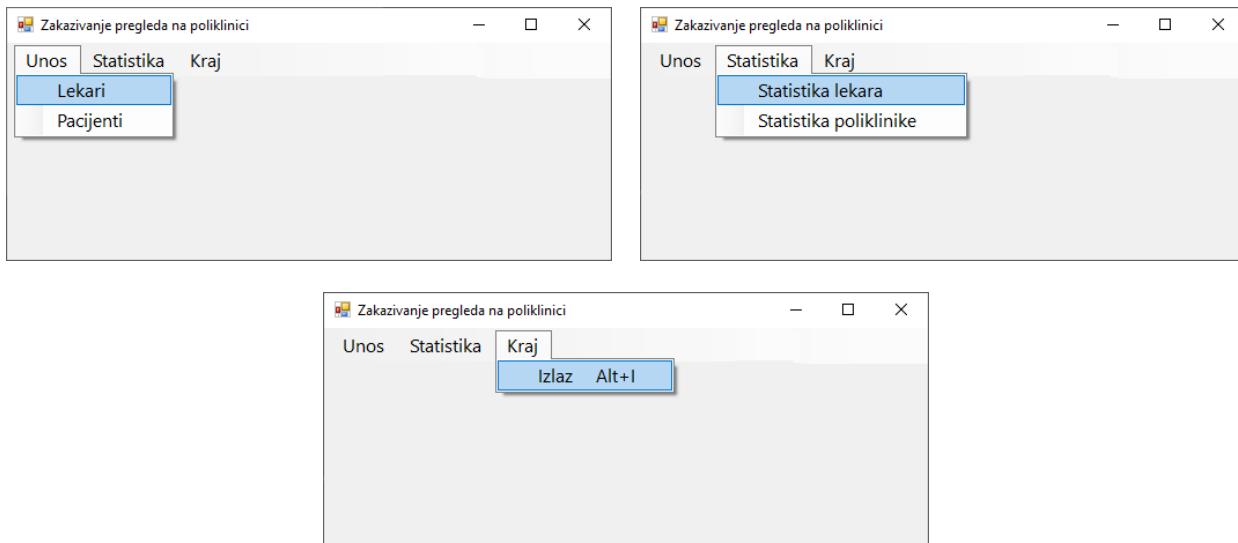


Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

За сваког лекара додати податак имејл са провером унесености правилне адресе.

Поликлиника је одлучила да пошаље одређени број својих лекара на специјализацију. Сваки лекар који заврши специјализацију има специјалност која се карактерише својим називом и описом. Један лекар може да има више специјалности; једна специјалност је карактеристика више лекара. Када лекар постане специјалиста за неку област памти се датум специјализирања.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о лекарима обавља се преко обрасца чији је изглед дат на слици:

За тип се приказују шифре сортиране у растући низ.

Ако постојилекар са унетом шифром, приказати податке о њему.

Дугме **Izađi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Statistika lekara**, приказује списак шифара, имена и презимена лекара одређеног типа, који су имали више од наведеног броја заказаних прегледа у траженој календарској години, сортиран у опадајући низ по броју прегледа.

Образац треба да изгледа као на слици:

	Šifra	Ime i prezime	Tip	Broj pregleda
	66	Hana Bel...	dermatolo...	17
	183	Iva Filipović	dermatolo...	17
▶	87	Radmila ...	dermatolo...	16
	2	Danijela ...	dermatolo...	16
	57	Dunja Bel...	dermatolo...	14
	146	Milja Barj...	dermatolo...	14
	97	Aleksandar...	dermatolo...	13

Poliklinika	Grad	Broj pregleda
Alba poliklinika	Apatin	9
Vivalab	Beograd	4
Poliklinika dr Drašković	Novi Sad	3

За тип лекара приказују се називи типова сортирани у растући поредак.

Ако тип лекара није унет претраживање ће се вршити по свим типовима.

Резултати упита се приказују у контроли за табеларни приказ података.

Избором ставке из контролеза табеларни приказ података, у контролиза приказ података у облику листе приказује се листа назива поликлиника, градова из којих су те поликлинике и број заказаних прегледа на свакој од њих, за оног лекара који је изабран, сортиран по називу поликлинике у растући абецедни низ.

Дугме **Izađi** затвара отворен образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б23

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б23
Назив радног задатка	ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за тип лекара	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 34)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако шифра постоји	5	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Обезбеђена измена података у бази по спецификацији задатка	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за тип лекара	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 27)	ТАЧНО	НЕТАЧНО
Онемогућено уношење само позитивних целих бројева за број заказаних прегледа и годину	2	0
Приказани тражени подаци у контроли за табеларни приказ података	10	0
Приказани тражени подаци у контроли за приказ података у облику листе	11	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

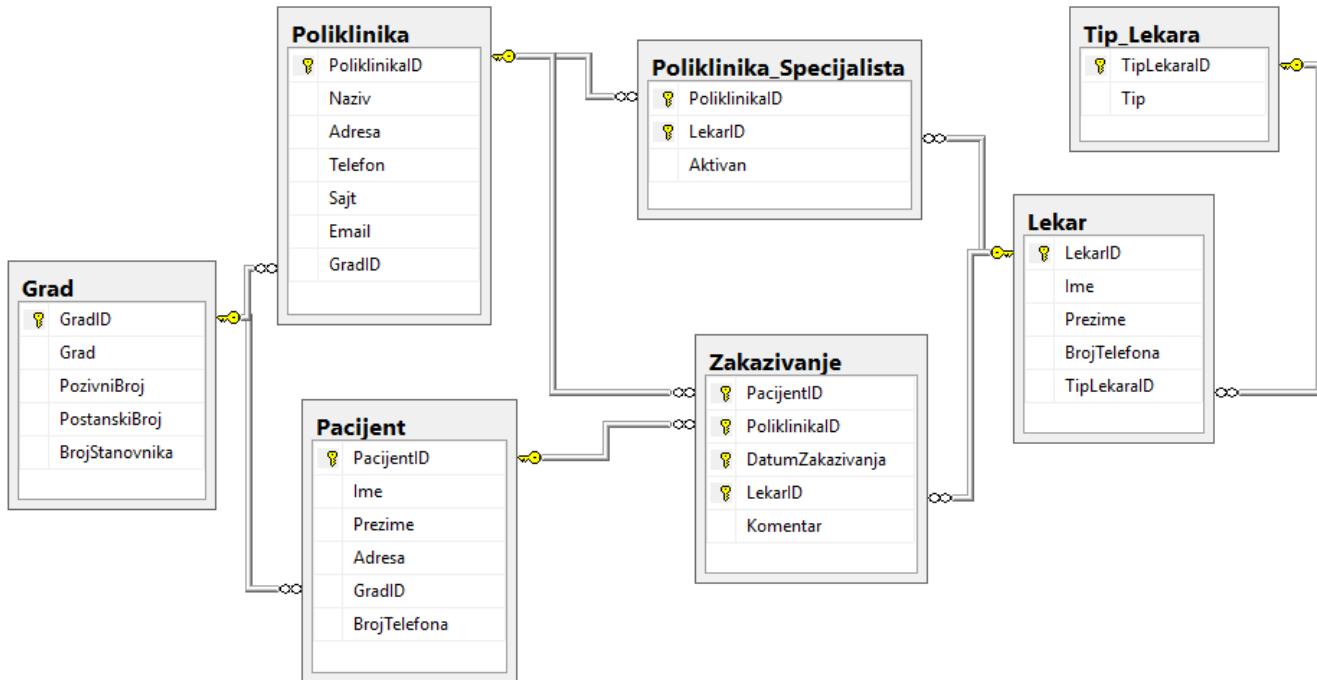
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б24

НАЗИВ РАДНОГ ЗАДАТКА: ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ

ЕИТ-Б24: ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ

Креирана је база података у којој се води евиденција о пацијентима који обављају прегледе на поликлиникама и њиховим лекарима.

Дијаграм базе је дат на слици:

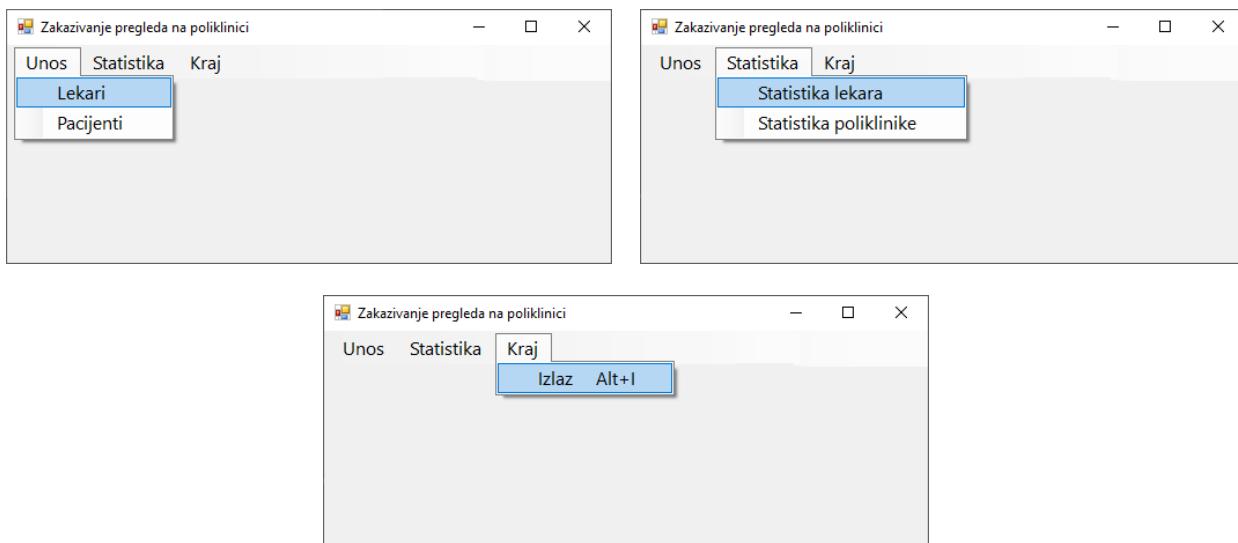


Користећи DDL наредбе, проширити базу тако да се задовоље следећи захтеви:

За сваког лекара додати податак имејл са провером унесености правилне адресе.

Поликлиника је одлучила да пошаље одређени број својих лекара на специјализацију. Сваки лекар који заврши специјализацију има специјалност која се карактерише својим називом и описом. Један лекар може да има више специјалности; једна специјалност је карактеристика више лекара. Када лекар постане специјалиста за неку област памти се датум специјализирања.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о пацијентима обавља се преко обрасца чији је изглед дат на слици:

За град се приказује назив града сортирани у растући низ.

Ако постоји пациент са унетом шифром, приказати податке о њему.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Statistika poliklinike**, приказује листу назива, адреса и градова поликлиника, укупан број заказаних прегледа у одређеном периоду оних пацијената који живе или не у истом граду у коме се налази поликлиника, сортирану по шифри поликлинике у растући низ.

Образац треба да изгледа као на слици:

	Šifra	Naziv	Adresa	Grad	Svi	Iz istog grada	Iz razlicit grada
▶	34	Magna ...	Bul. N...	Negotin	339	25	314
	35	Dr Djoric	Jovan...	Vranje	327	15	312
	36	Vivalab	Subot...	Beograd	368	24	344
	37	Poliklini...	Palih ...	Vlasotince	327	19	308
	38	Demetra	Rava...	Subotica	389	12	377
	39	Life Im...	Bul. N...	Novi Pazar	359	53	306
	40	Mediko...	Bul. 1...	Niš	365	22	343

Подаци се приказују у контролиза табеларни приказ података.

Избором ставке из контролеза табеларни приказ података, у контролиза приказ података у облику листе приказује се листа имена и презимена пацијената, град из кога долазе и датум када су заказали преглед на одабраној поликлиници, за тражени период, сортиран по датуму заказивања поликлинике у опадајући низ.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б24

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б24
Назив радног задатка	ЗАКАЗИВАЊЕ ПРЕГЛЕДА НА ПОЛИКЛИНИЦИ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за град	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 34)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако шифра постоји	5	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Обезбеђена измена података у бази по спецификацији задатка	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 4)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0

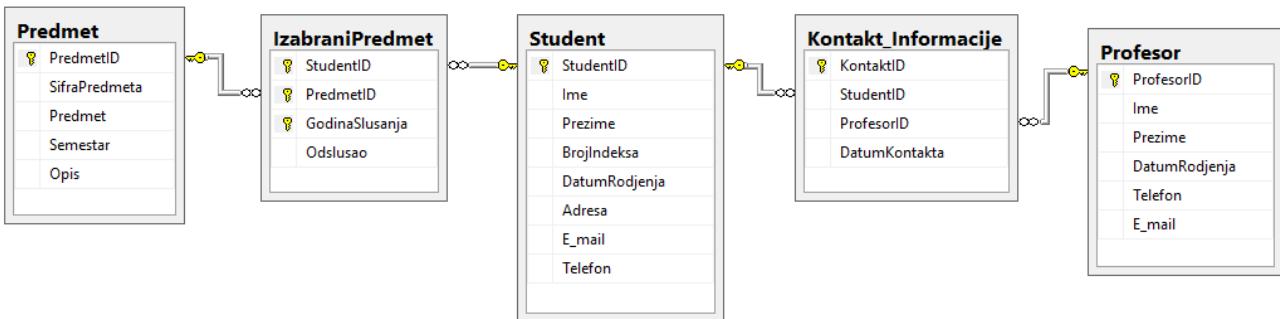
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Онемогућено уношење погрешног временског интервала	4	0
Приказани тражени подаци у контроли за табеларни приказ података	12	0
Приказани тражени подаци у контроли за приказ података у облику листе	12	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б25: КОМУНИКАЦИЈА СТУДЕНТА И ПРОФЕСОРА

За потребе факултета креирана је база података у којој се води евидентација о комуникацији професора и студената.

Дијаграм за постојећу базу дат је на слици:



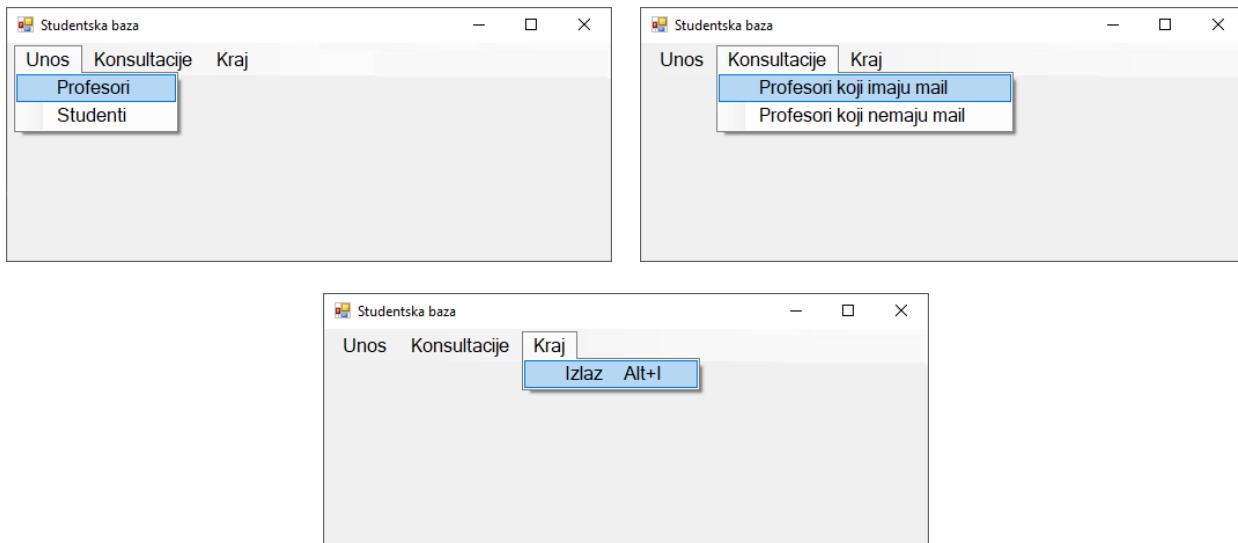
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

Када се оствари комуникација између професора и студента потребно је памтити и детаље комуникације (консултације, избор предмета за дипломски...).

Комуникација се може остварити на више начина због чега је потребно креирати нову табелу где ће се памтити тип контакта (мејл, телефон...)

Професори се током рада на факултету бирају у звања, због чега је потребно креирати нову табелу у којој би се памтила звања професора (доцент, ванредни професор, редовни професор...)

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о професорима обавља се преко обрасца чији је изглед дат на слици:

Šifra 202

Ime _____

Prezime _____

Datum rođenja _____

Telefon _____

Mail _____

Novi

Upiši

Izmjeni

Izadji

Lista unetih profesora:

- 195 - Emina Dukić
- 196 - Jerina Beloica
- 198 - Anita Andrejić
- 199 - Ljubinka Dukić
- 200 - Katarina Avramović
- 201 - Pera Perić

Šifra 12

Ime Dragica

Prezime Branković

Datum rođenja 04-04-1966

Telefon 067-349-6161

Mail dbrankovic@gmail.com

Novi

Upiši

Izmjeni

Izadji

Lista unetih profesora:

- 5 - Đurdina Avramović
- 6 - Uroš Filipović
- 7 - Lidija Andrić
- 8 - Savatije Dimitrijević
- 10 - Milja David
- 12 - Dragica Branković

Слогови се уписују тако што се приликом сваког новог уписа шифра аутоматски повећава за 1. При учитавању, форма је спремна за унос новог професора и у пољу шифре се приказује шифра под којом ће бити уписан нови професор. Исто се дешава и при клику на тастер **Novi**.

Уоквиру са листом приказују се подаци о професорима.

Уколико се нека ставка из оквира са листом изабере приказују се њени детаљи на обрасцу.

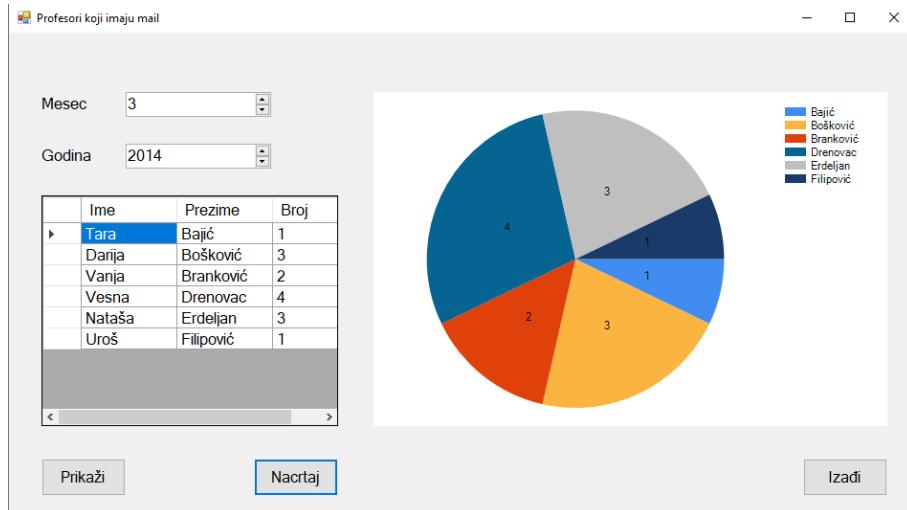
Обезбедити ажурирање оквира са листом после уноса и измене података.

Дугме **Izadji** затвара отворен образац и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Profesori koji imaju mail**, приказује листу имена, презимена и броја контакта са студентима у наведеном периоду, оних професора који имају мејл.

Образац треба да изгледа као на слици:



Резултате упита приказати у контролиза табеларни приказ података. На основу њих нацртати график броја контаката пронађених професора.

Дугме **Izadji** затвара отворен образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б25

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б25				
Назив радног задатка	КОМУНИКАЦИЈА СТУДЕНТА И ПРОФЕСОРА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за шифру	5	0
Постављена иницијална вредност у оквиру са листом	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 35)	ТАЧНО	НЕТАЧНО
Приказују се подаци за изабрану ставку оквира са листом	5	0
Обезбеђена функционалност дугмета за нови слог	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	8	0
Обезбеђено ажурирање података у бази по спецификацији задатка	8	0
Додата ставка у оквир са листом после уписа новог слога	3	0
Ажурирана ставка у оквир са листом после промене слога	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 4)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0

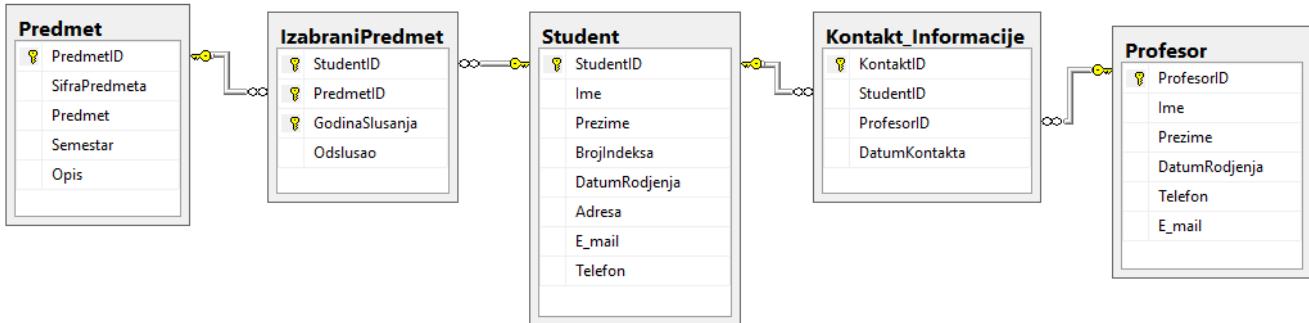
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 26)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	11	0
Нацртан график са траженим подацима	11	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б26: КОМУНИКАЦИЈА СТУДЕНТА И ПРОФЕСОРА

За потребе факултета креирана је база података у којој се води евидентација о комуникацији професора и студената.

Дијаграм за постојећу базу дат је на слици:



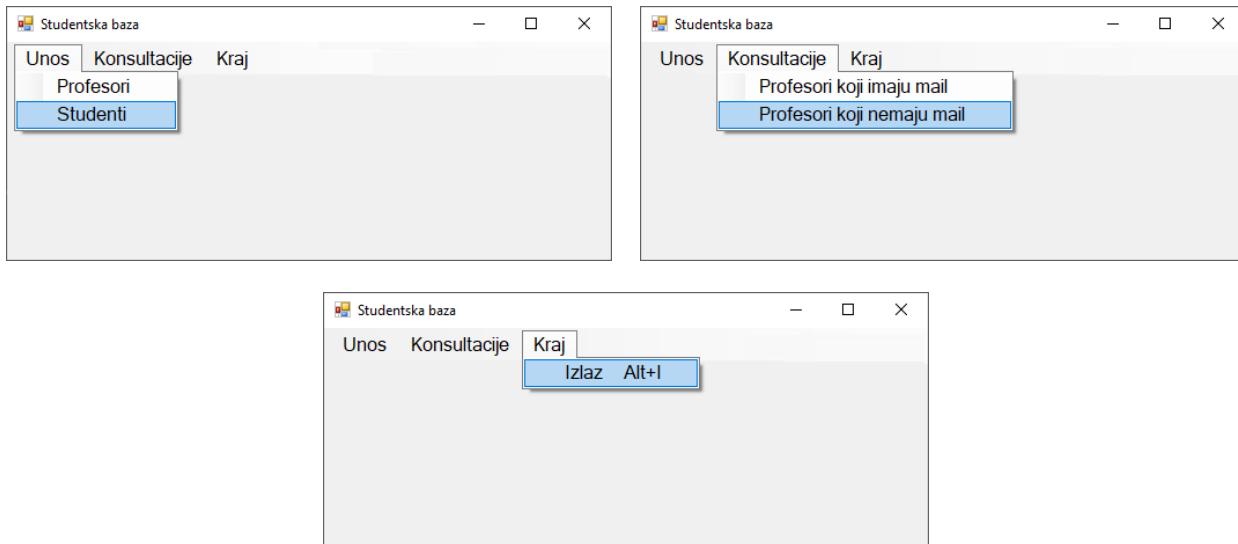
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

Када се оствари комуникација између професора и студента потребно је памтити и детаље комуникације (консултације, избор предмета за дипломски...).

Комуникација се може остварити на више начина због чега је потребно креирати нову табелу где ће се памтити тип контакта (мејл, телефон...)

Професори се током рада на факултету бирају у звања, због чега је потребно креирати нову табелу у којој би се памтила звања професора (доцент, ванредни професор, редовни професор...)

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о студентима обавља се преко обрасца чији је изглед дат на слици:

Šifra 502

Ime

Prezime

Broj indeksa

Datum rođenja

Adresa

Mail

Telefon

Izadji

Izmjeni

Upiši

Novi

Lista unetih studenata:

496 - Milan Babić
497 - Balša Hajdemann
498 - Nemanja Babić
499 - Milena Beločica
500 - Leon Aleksić
501 - Danica Mišić

Šifra 46

Ime Lidija

Prezime Dačić

Broj indeksa Y31110

Datum rođenja 16.11.1995

Adresa Beograd, Bul. Nikole Tesle 83

Mail lida@gmail.com

Telefon 064-133-3659

Izadji

Izmjeni

Upiši

Novi

Lista unetih studenata:

43 - Veljko Drenovac
44 - Rada Antić
45 - Hana Borovac
46 - Lidija Dačić
47 - Milan Bogavac
48 - Ema Dačić

Слогови се уписују тако што се приликом сваког новог уписа шифра аутоматски повећава за 1. При учитавању, форма је спремна за унос новог студента и у пољу шифре се приказује шифра под којом ће бити уписан нови студент. Исто се дешава и при клику на тастер **Novi**.

Уоквиру са листом приказују се подаци о студентима.

Уколико се нека ставка из оквира са листом изабере приказују се њени детаљи на обрасцу.

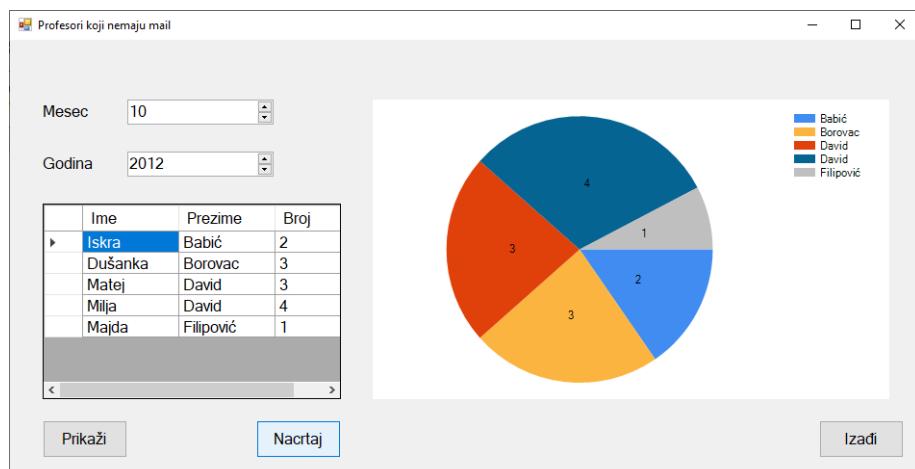
Обезбедите ажурирање оквира са листом.

Дугме **Izadji** затвара отворен образац и враћа на почетни мени.

Обезбедите да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама

Ставка менија **Profesori koji nemaju mail**, приказује листу имена, презимена и броја контакта са студентима у наведеном периоду, оних професора који немају мејл.

Образац треба да изгледа као на слици:



Резултате упита приказати у контролиза табеларни приказ података. На основу њих нацртати график броја контаката пронађених професора.

Дугме **Izadji** затвара отворен образац у и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б26

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б26
Назив радног задатка	КОМУНИКАЦИЈА СТУДЕНТА И ПРОФЕСОРА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за шифру	5	0
Постављена иницијална вредност у оквир са листом	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 35)	ТАЧНО	НЕТАЧНО
Приказују се подаци за изабрану ставку оквира са листом	5	0
Обезбеђена функционалност дугмета за нови слог	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	8	0
Обезбеђено ажурирање података у бази по спецификацији задатка	8	0
Додата ставка у оквир са листом после уписа новог слога у базу	3	0
Ажурирана ставка у оквир са листом после промене слога	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 4)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 26)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	11	0
Нацртан график са траженим подацима	11	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

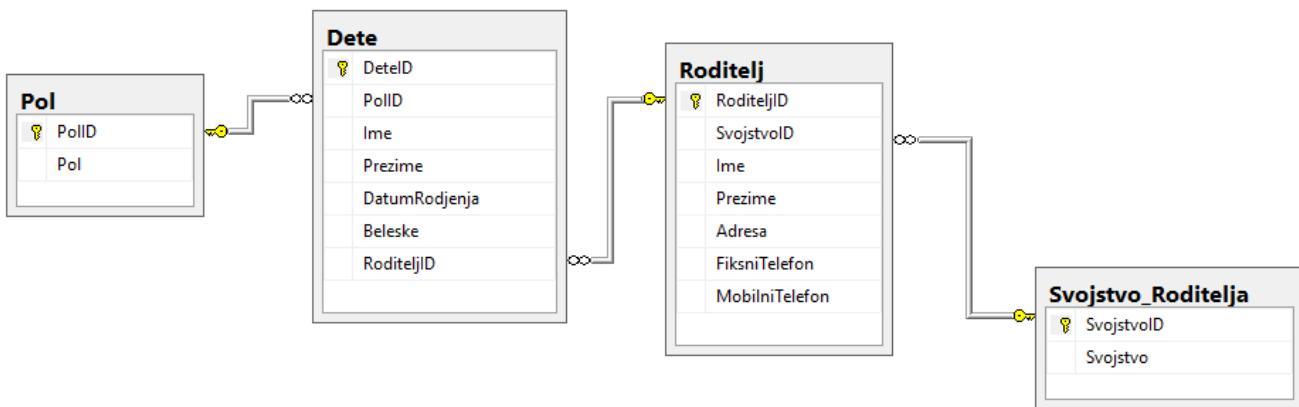
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б27

НАЗИВ РАДНОГ ЗАДАТКА: ДНЕВНИ БОРАВАК ДЕЦЕ

ЕИТ-Б27: ДНЕВНИ БОРАВАК ДЕЦЕ

За потребе приватног вртића креирана је база дневни боравак деце.

Дијаграм за постојећу базу дат је на слици:



Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

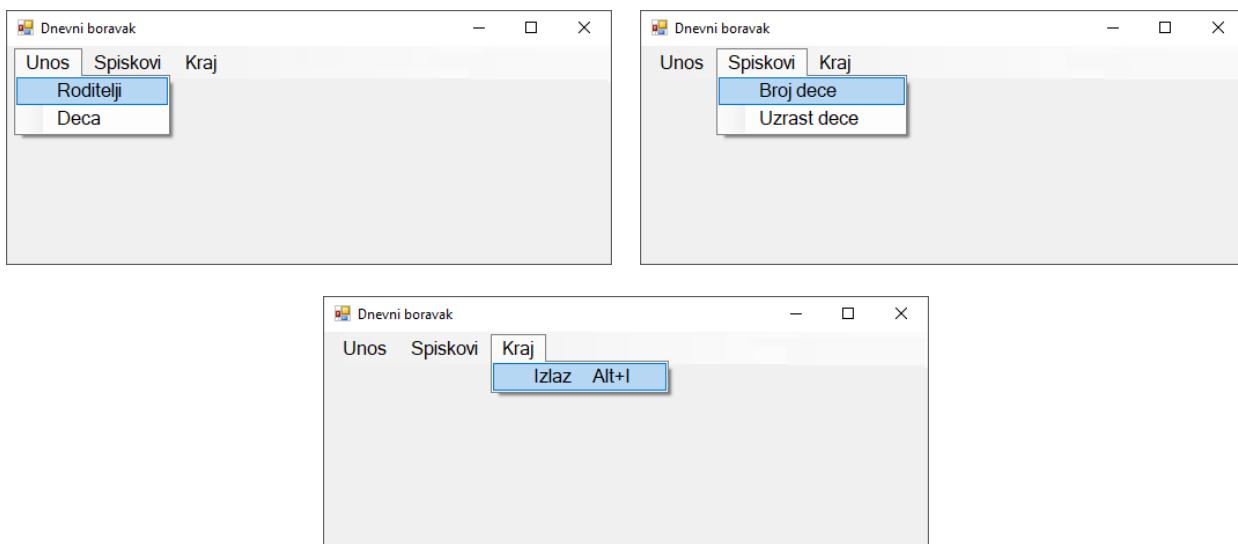
Обзиром да родитељи због својих обавеза немају доволно времена да подстакну различита интересовања код деце желимо да им понудимо што више различитих активности.

Креирати табелу активност у којој ће се чувати назив активности и време почетка и завршетка активности.

Активности се одвијају увек истим даном, а у току једног дана дете може имати више различитих активности.

У посебном регистру активности воде се подаци о датуму активности, да ли је дете било присутно или не и белешка о детету и његовом напредовању.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о родитељима обавља се преко обрасца чији је изглед дат на слици:

This screenshot shows a Windows application window titled 'Roditelji'. The form contains fields for 'Šifra' (1), 'Ime' (Dušanka), 'Prezime' (Aleksić), 'Adresa' (Zaječar, Božidara Adžije 36/1), 'Fiksni telefon' (021.021-0106), 'Mobilni telefon' ((068) 651-0147), and 'Svojstvo' (radio buttons for otac, majka, staratelj). Below the form are buttons: 'Novi' (New), 'Unesi' (Enter), 'Prethodni' (Previous), 'Sledeći' (Next), 'Nacrtaj' (Draw), and 'Izadi' (Exit).

This screenshot shows the same 'Roditelji' application window, but with several red annotations: a blue circle is drawn over the 'Ime' field containing 'Ana', and a green rectangle is drawn over the 'Prethodni' button.

На почетку су сва поља празна. Када је поље шифре празно, клик на дугме **Prethodni** приказује првог родитеља из листе, док клик на **Sledeći** пребације на крај и приказује последњег у листи.

Притиском на дугме **Novi** обрише се садржај свих поља и припреми се упис новог родитеља у базу.

Притиском на дугме **Nacrtaj** црта се чича Глиша, на начин који је дат на слици.

Обезбедити да дугме **Nacrtaj** може да се притисне само једном.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису података, као и о обрађеним грешкама.

Ставка менија **Broj dece**, приказује листу имена, презимена, адресе родитеља који су у траженом својству и имају одређен број деце одређеног пола, сортирану у растући абецедни низ по имену, а затим по презимену.

Образац треба да изгледа као на слици:

This screenshot shows a Windows application window titled 'Broj Dece'. It has two sections: 'Svojstvo' (checkboxes for Majka, Otac, Staratelj) and 'Pol' (checkboxes for Muški, Ženski). To the right is a dropdown for 'Broj dece' (2) and buttons 'Prikaži' (Show) and 'Izadi' (Exit). Below these is a table listing children:

	Ime	Prezime	Adresa	Svojstvo	Broj dece
Danijal	Bebić	Negotin, Gornjo Mateje...	Otac	2	
Kaja	Hajdeman	Ruma, Braće Tasković...	Majka	2	
Lena	Filipović	Sombor, Vazduhoplova...	Majka	2	
Milan	Apostolović	Bećej, Sicevačka 46	Otac	2	
Miloš	Bajić	Vranje, Bul. Mihajla Pu...	Otac	2	

Минимални број деце је 1, а максимални 6.

Подаци се приказују у контроли за табеларни приказ података.

Дугме **Izadji** затвара отворен образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б27

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б27				
Назив радног задатка	ДНЕВНИ БОРАВАК ДЕЦЕ				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 7)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 43)	ТАЧНО	НЕТАЧНО
Обезбеђена функционалност дугмета за унос новог слога	3	0
Обезбеђено уписивање података у базу по спецификацији задатка	10	0
Обезбеђена функционалност дугмета за приказ следећег слога	8	0
Обезбеђена функционалност дугмета за приказ предходног слога	8	0
Нацртана слика	10	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 6)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за број деце	2	0

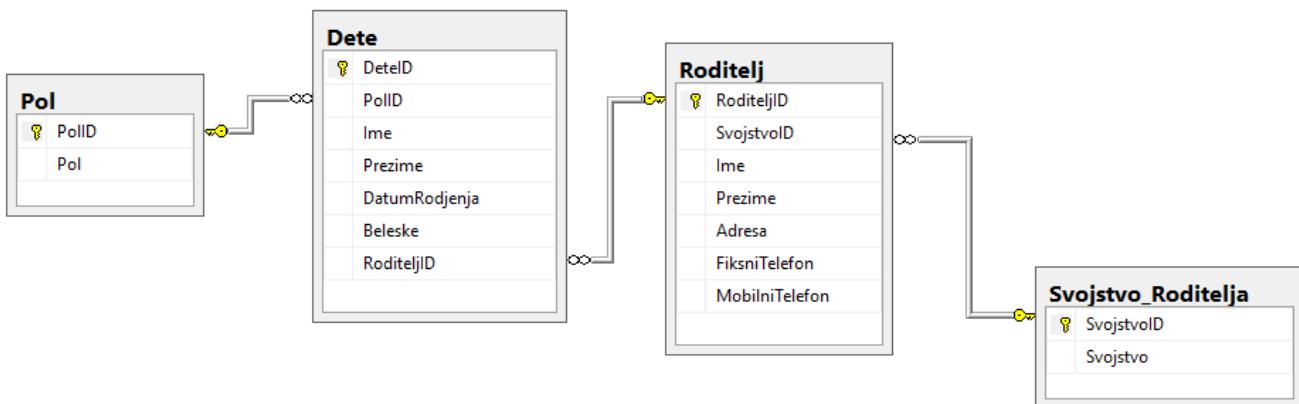
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б28: ДНЕВНИ БОРАВАК ДЕЦЕ

За потребе приватног вртића креирана је база дневни боравак деце.

Дијаграм за постојећу базу дат је на слици:



Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

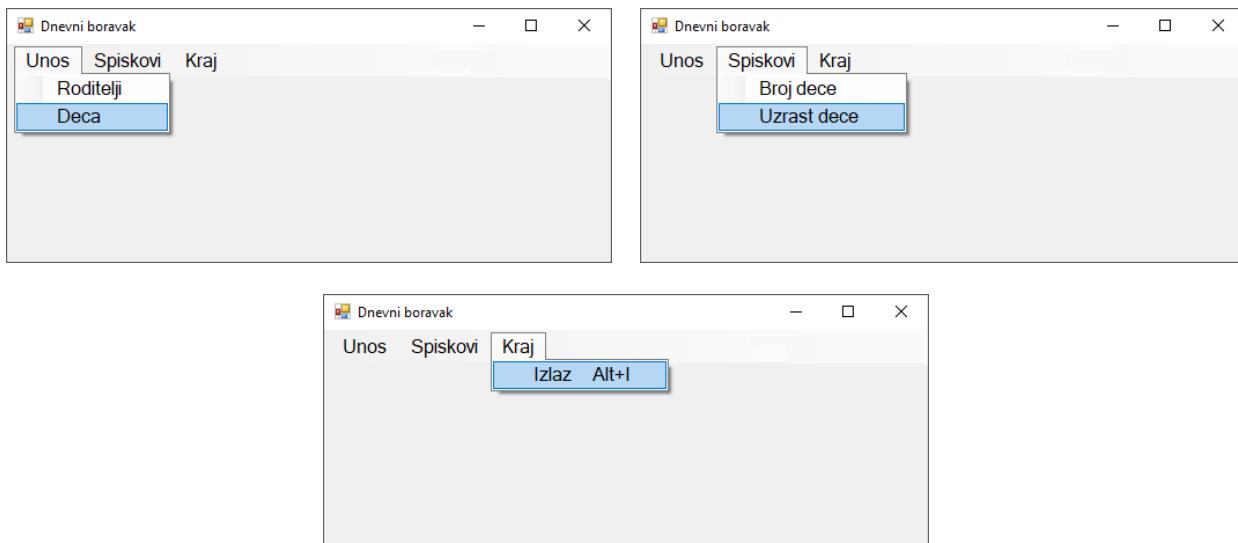
Обзиром да родитељи због својих обавеза немају доволно времена да подстакну различита интересовања код деце желимо да им понудимо што више различитих активности.

Креирати табелу активност у којој ће се чувати назив активности и време почетка и завршетка активности.

Активности се одвијају увек истим даном, а у току једног дана може имати више различитих активности.

У посебном регистру активности воде се подаци о датуму активности, да ли је дете било присутно или не и белешка о детету и његовом напредовању.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о деци обавља се преко обрасца чији је изглед дат на слици:

На почетку су сва поља празна. Када је поље шифре празно, клик на дугме **Prethodni** приказује прво дете из листе, док клик на **Sledeći** пребацује на крај и приказује последње дете у листи.

Притиском на дугме **Novi** обрише се садржај свих поља и припреми се упис новог детета у базу.

Притиском на дугме **Nacrtaj** црта се чича Глиша, на начин који је дат на слици

Обезбедити да дугме **Nacrtaj** може да се притисне само једном.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису података, као и о обрађеним грешкама.

Ставка менија **Uzrast dece**, приказује листу имена, презимена, датума рођења и пола, оне деце која су одређеног узраста и одређеног пола.

Образац треба да изгледа као на слици:

	Ime	Prezime	DatumRodjenja	Pol
Dobrla	Dačić	3/10/2012	Zenski	
Darija	Bajić	5/27/2012	Zenski	
Žana	Aleksić	4/19/2012	Zenski	
Zorica	Erceg	1/12/2012	Zenski	
Brankica	Erceg	11/22/2012	Zenski	
Ivona	Bajić	9/30/2012	Zenski	
Vesna	Branković	4/1/2012	Zenski	

Минимална вредност за узраст је 0, а максимална 7.

Подаци се приказују контролиза табеларни приказ података.

Дугме **Izadji** затвара отворен образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б28

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б28
Назив радног задатка	ДНЕВНИ БОРАВАК ДЕЦЕ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 7)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креирана форма за унос података са траженим елементима	4	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 43)	ТАЧНО	НЕТАЧНО
Обезбеђена функционалност дугмета за унос новог слова	3	0
Обезбеђено уписивање података у базу по спецификацији задатка	10	0
Обезбеђена функционалност дугмета за приказ следећег слова	8	0
Обезбеђена функционалност дугмета за приказ предходног слова	8	0
Нацртана слика	10	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 6)	ТАЧНО	НЕТАЧНО
Креирана форма са траженим елементима	4	0
Постављена иницијална вредност у оквир за текст са бројачем	2	

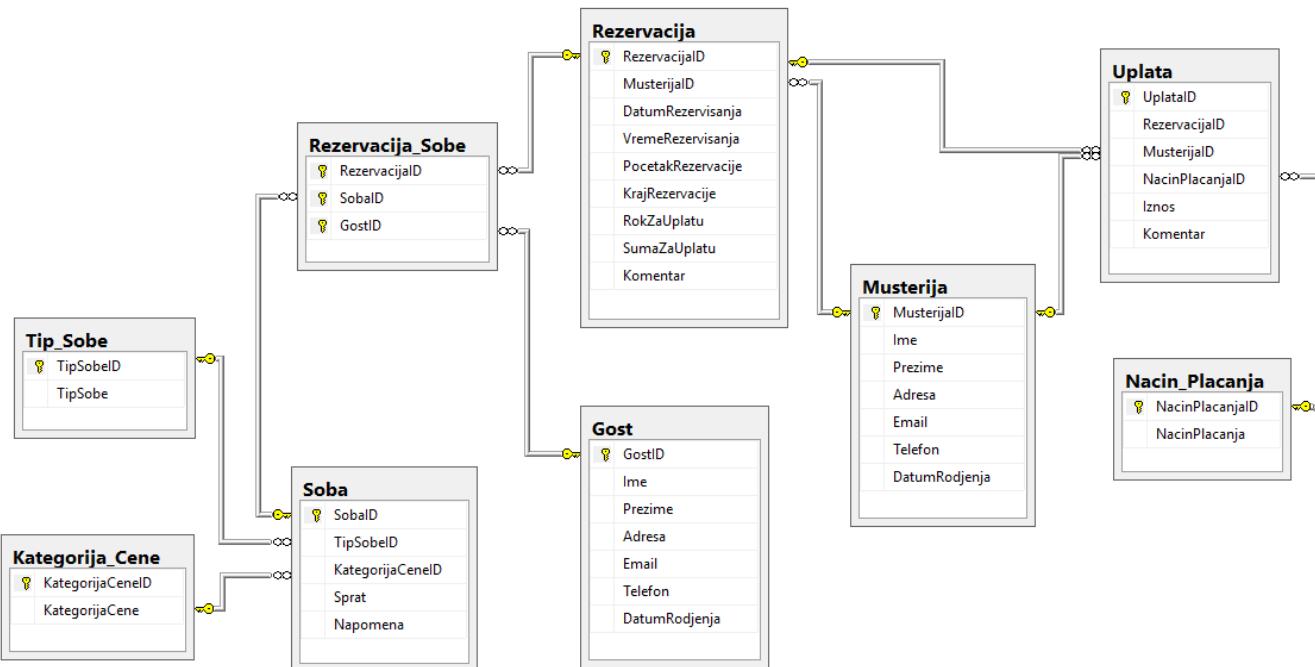
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б29: РЕЗЕРВАЦИЈА ХОТЕЛСКИХ СОБА

За потребе хотела који жели да води евиденцију о резервацији хотелских соба креирана је база података.

Дијаграм за постојећу базу дат је на слици:



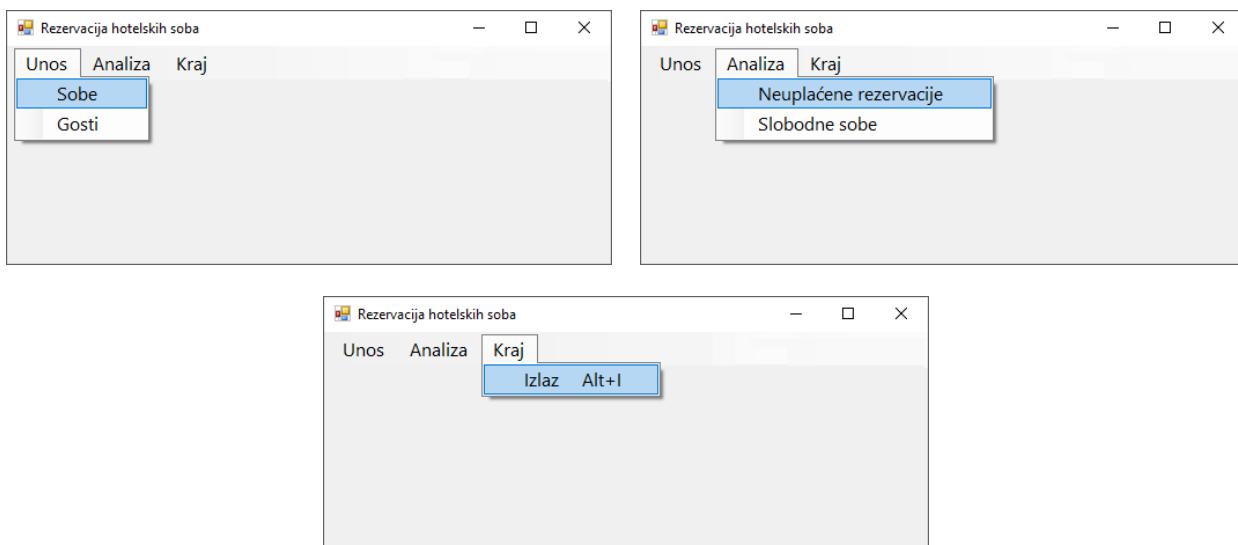
Објашњење шеме: муштерија је онај који прави резервацију и на чије име се врши уплата (он резервише и плаћа услугу, али не мора бити и корисник услуге). Муштерија може једном резервацијом да резервише више соба на више имена, тј. за више различитих особа (гости – корисници услуге).

Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

За резервацију додати поље логичког типа које каже да ли је уплата извршена или не.

Хотел жели да подигне квалитет својих услуга и да понуди за сваку собу додатне погодности (интернет, телевизор, мини бар...). Обзиром да у различитим собама постоји могућност више додатака креирати и додатну табелу у којој ће се чувати детаљи о сваком додатку (брзина протока интернета, тип телевизора...)

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о собама обавља се преко обрасца чији је изглед дат на слици:

Soba	Tip	Kategorija	Srat
1	jednokrevetna soba 1/1	3* zvezdice	1
2	trokrevetni app 1/3	Prva kategorija	10
3	trokrevetna soba 1/3	Prva kategorija	3
4	četvorokrevetni duplex a...	Prva kategorija	2
5	trokrevetna soba 1/3	2* zvezdice	9
6	trokrevetna soba 1/3	Druga kateg...	3
7	dvokrevetna soba 1/2	4* zvezdice	10
8	jednokrevetna soba 1/1	2* zvezdice	1
9	trokrevetni app 1/3	Prva kategorija	2
10	trokrevetni app 1/3	Druga kateg...	6
11	dvokrevetna soba 1/2	Druga kateg...	4
12	trokrevetna soba 1/3	4* zvezdice	10

За тип се приказују називи типова сортирани у растући низ.

За категорију се приказују називи категорија сортирани у растући низ.

Подаци о унетим собама приказују се у контроли за приказ података у облику листе.

Уколико соба са унетом ознаком постоји, приказују се подаци о њој.

Обезбедити ажурирање контроле за приказ података у облику листе.

Дугме **Izadjii** затвара отворен образац и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Neuplaćene rezervacije**, приказује листу неуплаћених резервација до данашњег датума.

Образац треба да изгледа као на слици:

	Rezervacija	Šifra	Ime	Prezime	Ukupan iznos	Plaćeno
▶	2	246	Aleksandar	Borovac	2100	
	3	231	Radmila	Dačić	390	200
	7	272	Damjan	Beloica	3040	2739
	15	295	Nataša	Filipović	4250	3847
	16	10	Milja	David	1840	1539
	20	95	Katarina	Barjaktare...	2400	2199
	25	210	Una	Bebić	10300	9699

У контроли за табеларни приказ података приказује се листа идентификационих бројева резервација, шифра, име и презиме госта који по њој одседа у хотелу, укупна сума за уплату, уплаћен износ по резервацији за све резервације које нису у потпуности уплаћене до данашњег датума, сортиране по идентификационом броју резервације у растући низ.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б29

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б29				
Назив радног задатка	РЕЗЕРВАЦИЈА ХОТЕЛСКИХ СОБА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 21)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за тип собе	5	0
Постављена иницијална вредност за категорију	5	0
Постављена иницијална вредност у контроли за приказ података у облику листе	4	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 36)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако унета шифра постоји	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	8	0
Додата ставка у контролу за приказ података у облику листе после уписа слога у бази	2	0
Обезбеђено брисање података из базе по спецификацији задатка	6	0
Обрисана ставка из контролеза приказ података у облику листе после брисања слога из базе	2	0
Обезбеђена измена података из базе по спецификацији задатка	8	0
Измењена ставка у контроли за приказ података у облику листе после ажурирања слога у бази	2	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 6)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за данашњи датум	2	0

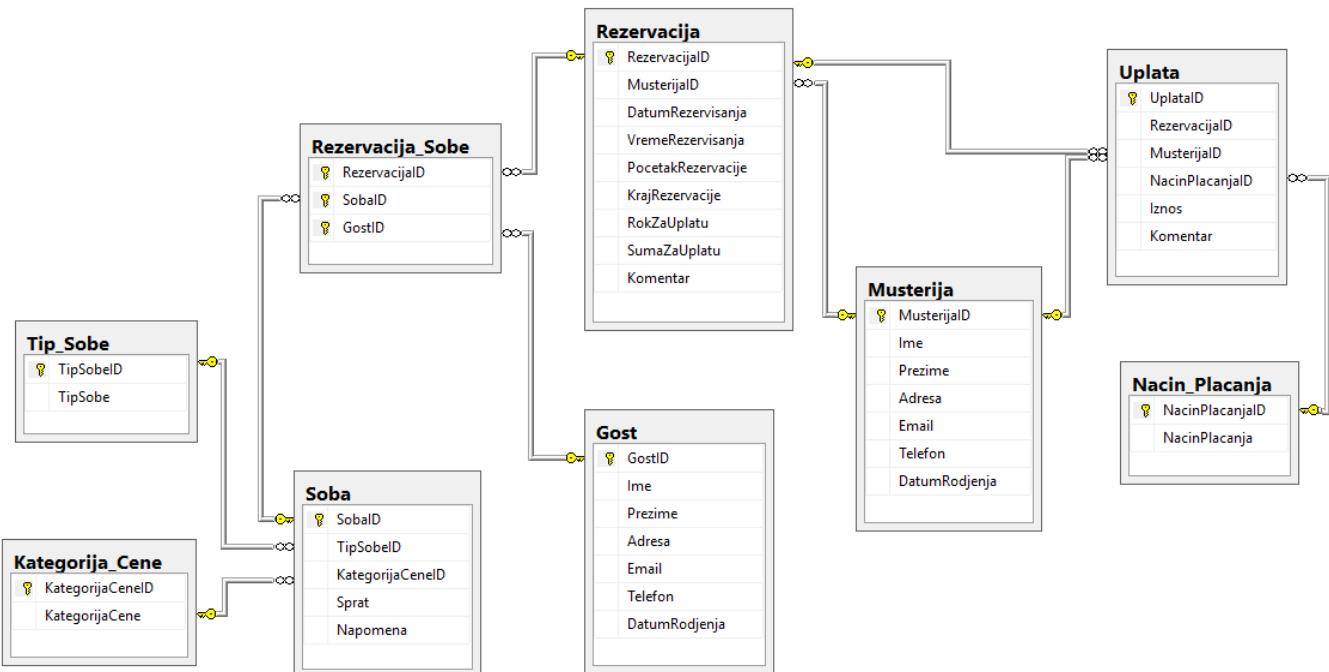
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 19)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	15	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б30: РЕЗЕРВАЦИЈА ХОТЕЛСКИХ СОБА

За потребе хотела који жели да води евиденцију о резервацији хотелских соба креирана је база података.

Дијаграм за постојећу базу дат је на слици:



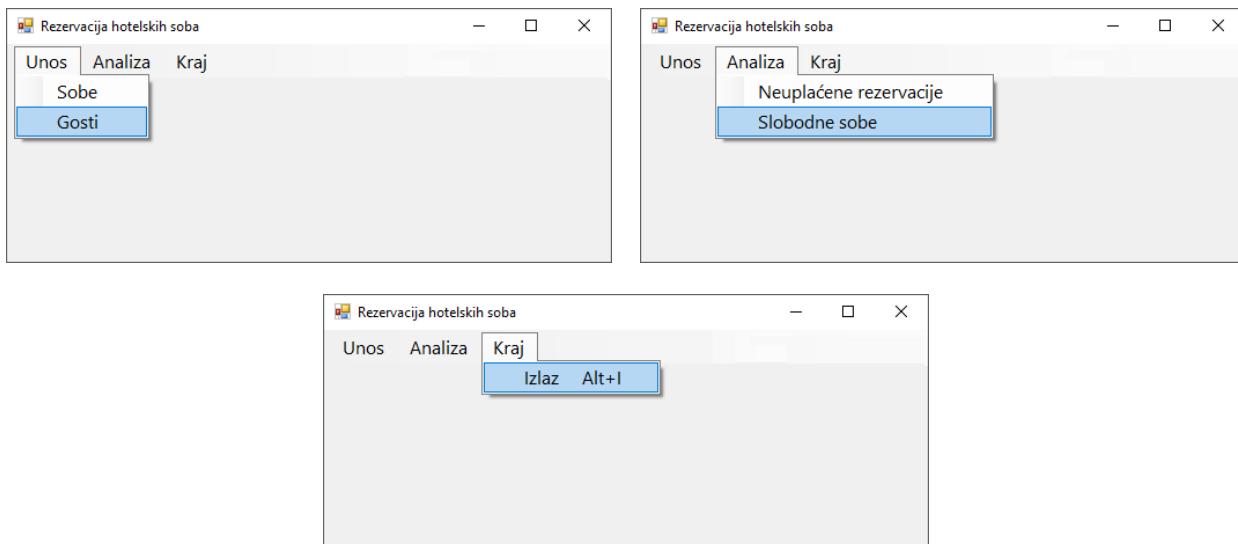
Објашњење шеме: муштерија је онај ко прави резервацију и на чије име се врши уплата (он резервише и плаћа услугу, али не мора бити и корисник услуге). Муштерија може једном резервацијом да резервише више соба на више имена, тј. за више различитих особа (гости – корисници услуге).

Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

За резервацију додати поље логичког типа које каже да ли је уплата извршена или не.

Хотел жели да подигне квалитет својих услуга и да понуди за сваку собу додатне погодности (интернет, телевизор, мини бар...). Обзиром да у различитим собама постоји могућност више додатака креирати и додатну табелу у којој ће се чувати детаљи о сваком додатку (брзина протока интернета, тип телевизора...)

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о гостима обавља сепреко обрасца чији је изглед дат на слици:

Šifra	Ime	Prezime	Adresa	E mail	Telefon	Datum rođenja
7	Nikša	Dimitrijević	Pančevo, Braće Tasković...	kdrnts.klunv...	068/641-9...	22.10.1990
8	Zorica	Andrejić	Vranje, Bul. Nikole Tesle ...	mtnw8@yahoo...	068/288-7...	
9	Ljiljana	Erdejan	Valjevo, Gornjo Matejeva...	yupxez.hazx...	069/105-0...	2.7.1972
10	Uroš	Dimitrijević	Paraćin, Partizanska 68/2	qbpe@gmail...	068/095-5...	16.1.1957
11	Milja	Andrejić	Niš, Bul. Mihajla Pupina 6	tqditu6@hot...	062/763-5...	4.8.1972
12	Ivana	Bajić	Vlasotince, Lala 6		068/101-2...	10.12.1959

У контроли за приказ података у облику листе приказују се подаци о унетим гостима.

Уколико гост са унетом шифром постоји, приказују се подаци о њему.

Обезбедити ажурирање контроле за приказ података у облику листе.

Дугме **Izadji** затвара отворен образац и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Slobodne sobe**, приказује листу слободних соба, у односу на датум, тип и категорију.

Образац треба да изгледа као на слици:

SobalID	TipSobe	KategorijaCene	Sprat
12	trokrevetna soba 1/3	4* zvezdice	10
146	trokrevetna soba 1/3	4* zvezdice	8
157	trokrevetna soba 1/3	4* zvezdice	7

За тип собе као прву ставку приказује се реч „– sve –“, а после тога називи типова соба.

За категорију као прву ставку приказује се реч „– sve –“, а после тога називи категорија.

У контроли за табеларни приказ података приказује се листа ознаке собе, назив типа, категорија и спрат, за оне собе које су слободне на тражени дан, за одабран тип собе и категорију.

Дугме **Izadji** затвара отворен образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б30

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б30				
Назив радног задатка	РЕЗЕРВАЦИЈА ХОТЕЛСКИХ СОБА				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност у контроли за приказ података у облику листе	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 36)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља ако унета шифра постоји	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	8	0
Додата ставка у контролуза приказ података у облику листе после уписа слога у бази	2	0
Обезбеђено брисање података из базе по спецификацији задатка	6	0
Обрисана ставка из контролеза приказ података у облику листе после брисања слога из базе	2	0
Обезбеђена измена података из базе по спецификацији задатка	8	0
Измењена ставка у контроли за приказ података у облику листе после ажурирања слога у бази	2	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 16)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за данашњи датум	2	0
Постављена иницијална вредност за тип собе	5	0
Постављена иницијална вредност за категорију	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	14	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

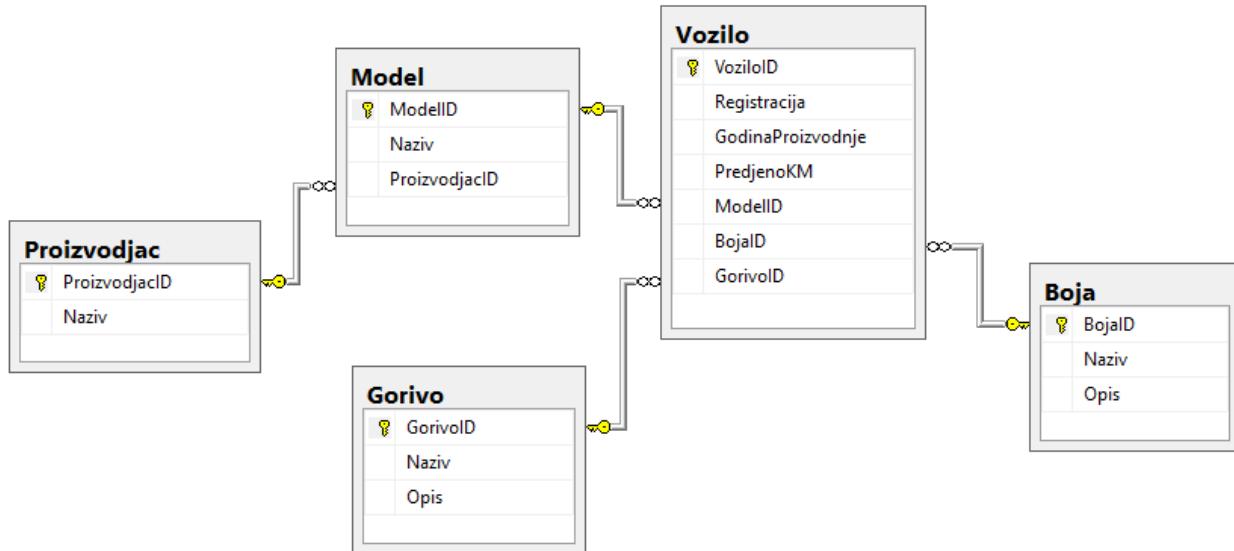
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б31

НАЗИВ РАДНОГ ЗАДАТКА: АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА

ЕИТ-Б31: АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА

За потребе ауто плаца креирана је база података у којој се води евиденција о свим возилима.

Дијаграм за постојећу базу дат је на слици:



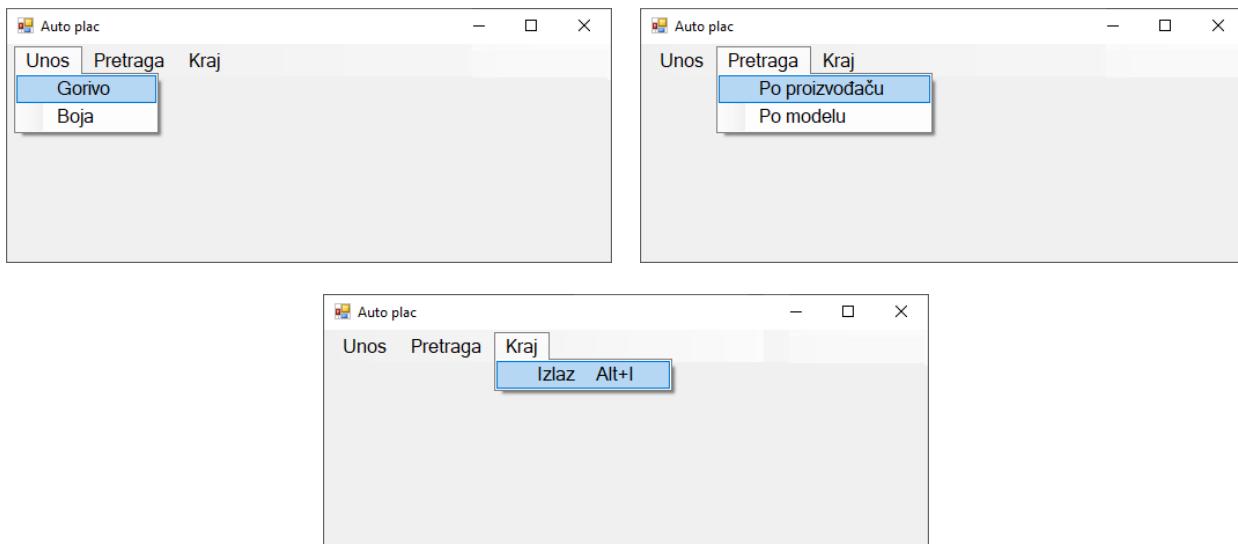
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

У табели возило желимо дачувамо цену возила у тренутку куповине.

Наша возила имају различите елементе опреме. Желимо да за сваки елемент опреме чувамо назив опреме (клима, подизачи стакала, ваздушни јастуци,...) и детаљан опис (мануелна или аутоматска клима, број ваздушних јастука, абс, есп,...).

Обзиром да различита возила могу имати различите елементе опреме везу више на више између ове две табеле трансформисати у две везе један на више увођењем нове табеле.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о гориву обавља се преко обрасца чији је изглед дат на слици наставку.

У оквиру са листом приказују се подаци о гориву сортирани по шифри у растући низ.

Уколико гориво са унетом шифром постоји приказују се подаци о њему.

Обезбедити ажурирање оквира са листом.

Šifra	Naziv	Opis
1	Evrodizel	
2	Evro BMB 98	
3	TNG auto-gas	specifikacija br.
4	EUROSUPER 95	
5	SUPER 95	specifikacija br.
6	Evropremium BMB 95	specifikacija br.
7	Nafta	za autobuse

Дугме **Izadj**i затвара отворен образац и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Po proizvodjacu**, приказује регистрације возила и годину њихове производње, за одређеног произвођача, боју, врсту горива и опсег пређених километара, сортираних по години производње у опадајући, а по регистрацији у растући низ.

Образац треба да изгледа као на слици:

Registracija	GodinaProizvodnje
SD-864-EY	2013
TO-188-TM	2004
HL-082-PX	1995
JR-670-TD	1995

За произвођача се приказују називи производа сортирани у растући низ.

За боју се приказују називи боја сортирани у растући низ.

За врсту горива се приказују називи горива сортирани у растући низ.

За километражу се приказују следећи подаци:

- 0 – 30000 km
- 30001 – 60000 km
- 60001 – 90000 km
- 90001 – 120000km
- 120001 – 150000km
- 150001 – 180000km
- 180001 – 210000 km
- preko 210000km

Резултати упита приказују се у контролиза табеларни приказ података.

Дугме **Izadj**i затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б31

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б31
Назив радног задатка	АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИИ

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност у оквир са листом	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Обезбеђено уписивање података у базу по спецификацији задатка	7	0
Додата ставка у оквир са листом после уписа слога у бази	2	0
Обезбеђено брисање података из базе по спецификацији задатка	6	0
Обрисана ставка из оквира са листом после брисања слога из базе	2	0
Обезбеђена измена података из базе по спецификацији задатка	8	0
Измењена ставка у оквиру са листом после ажурирања слога у бази	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за произвођача	5	0
Постављена иницијална вредност за боју	5	0
Постављена иницијална вредност за врсту горива	5	0
Постављена иницијална вредност за километражу	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 14)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	10	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

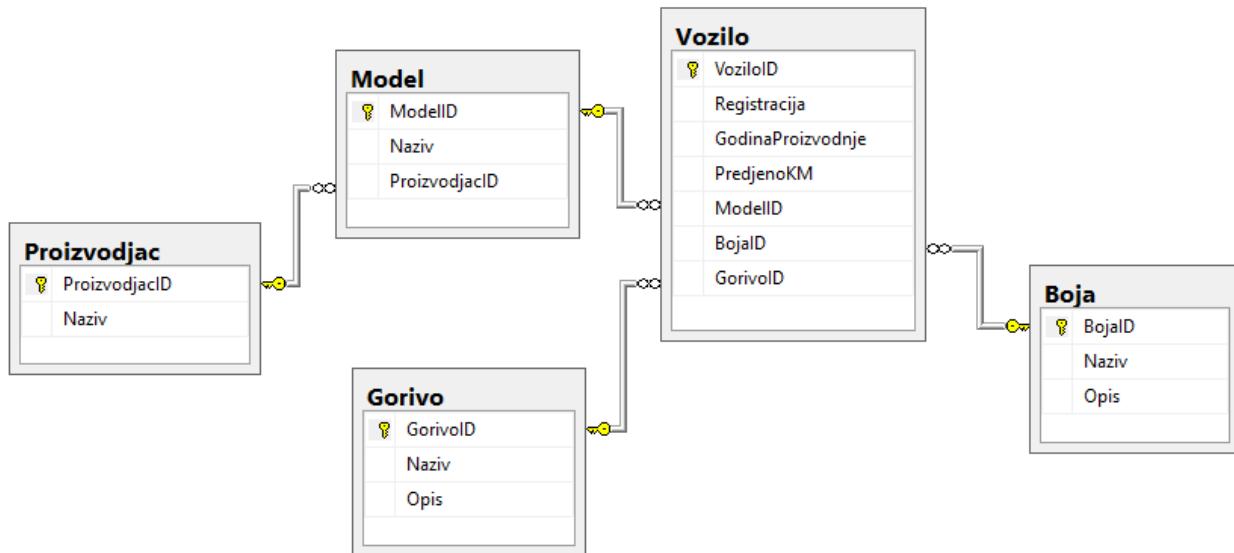
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б32

НАЗИВ РАДНОГ ЗАДАТКА: АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА

ЕИТ-Б32: АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА

За потребе ауто плаца креирана је база података у којој се води евиденција о свим возилима.

Дијаграм за постојећу базу дат је на слици:



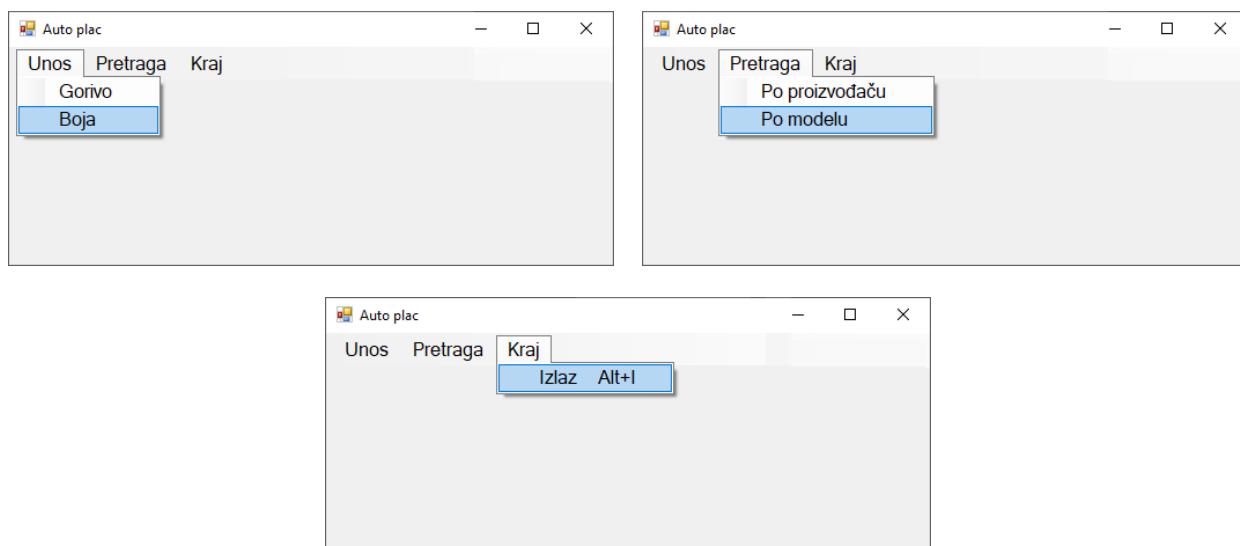
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

У табели возило желимо да чувамо цену возила у тренутку куповине. Цена треба да буде у динарима записана у следећем облику (нпр. 500,000.00 дин.).

Наша возила имају различите елементе опреме. Желимо да за сваки елемент опреме чувамо назив опреме (клима, подизачи стакала, ваздушни јастуци,...) и детаљан опис (мануелна или аутоматска клима, број ваздушних јастука, абс, есп,...).

Обзиром да различита возила могу имати различите елементе опреме везу више на више између ове две табеле трансформисати у две везе један на више увођењем нове табеле.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о боји обавља се преко обрасца чији је изглед дат на слици у наставку.

У оквиру са листом приказују се подаци о боји сортирани по шифри у растући низ.

Уколико унета шифра боје постоји, приказати одговарајуће податке.

Обезбедити ажурирање оквира са листом.

Šifra	Naziv	Opis
1	Crvena	opis Q-16
2	Tamno plava	opis Q-81
3	Zelena	opis K-23
4	Crna	
5	Orange	opis F-32
7	Žuta	opis P-70
9	Bela	opis W-69
10	Srebrna	

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Po modelu**, приказује регистрације возила и њихову пређену километражу, за одређен модел возила, боју, врсту горива и опсег године производње, сортираних по километражи у растући, а по регистрацији у опадајући низ.

Образац треба да изгледа као на слици:

Registracija	PredjenoKM
IL-006-MB	226780,25...
MG-966-IO	160863,36...
HM-104-LX	233079,58...
*	

За модел се приказује назив модела.

За боју се приказује назив боје.

За врсту горива се приказује назив горива.

За годину производње приказују се следећи подаци:

posle 2015
2010 - 2014
2005 - 2009
2000 - 2004
1995 – 1999
1990 – 1991
pre 1990

Резултати упита приказују се у контролиза табеларни приказ података.

Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б32

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б32
Назив радног задатка	АУТО ПЛАЦ СА ПОЛОВНИМ РЕГИСТРОВАНИМ ВОЗИЛИМА
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност у оквир са листом	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Обезбеђено уписивање података у базу по спецификацији задатка	7	0
Додата ставка у оквир са листом после уписа слога у базу	2	0
Обезбеђено брисање података из базе по спецификацији задатка	6	0
Обрисана ставка из оквира са листом после брисања слога из базе	2	0
Обезбеђена измена података из базе по спецификацији задатка	8	0
Измењена ставка у оквиру са листом после ажурирања слога у бази	3	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за модел	5	0
Постављена иницијална вредност за боју	5	0
Постављена иницијална вредност за врсту горива	5	0
Постављена иницијална вредност за годину производње	5	0

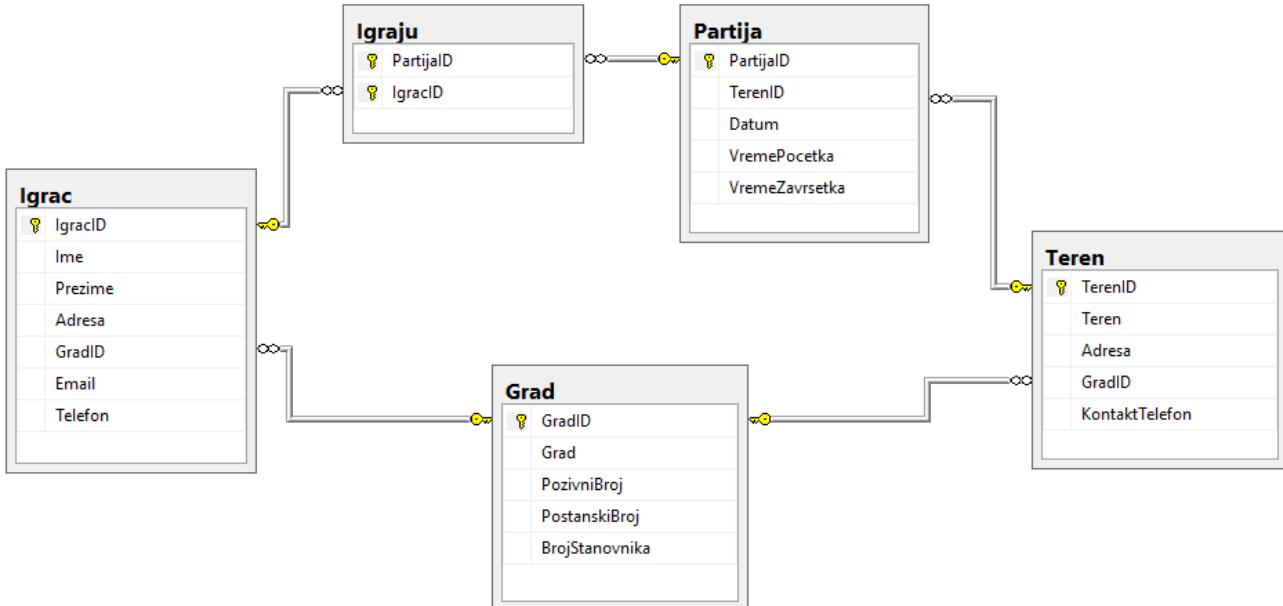
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 14)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	10	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б33: ГОЛФ КЛУБ

За потребе голф клуба креирана је база података у којој се води евиденција о члановима клуба, теренима и одиграним мечевима.

Дијаграм за постојећу базу дат је на слици:



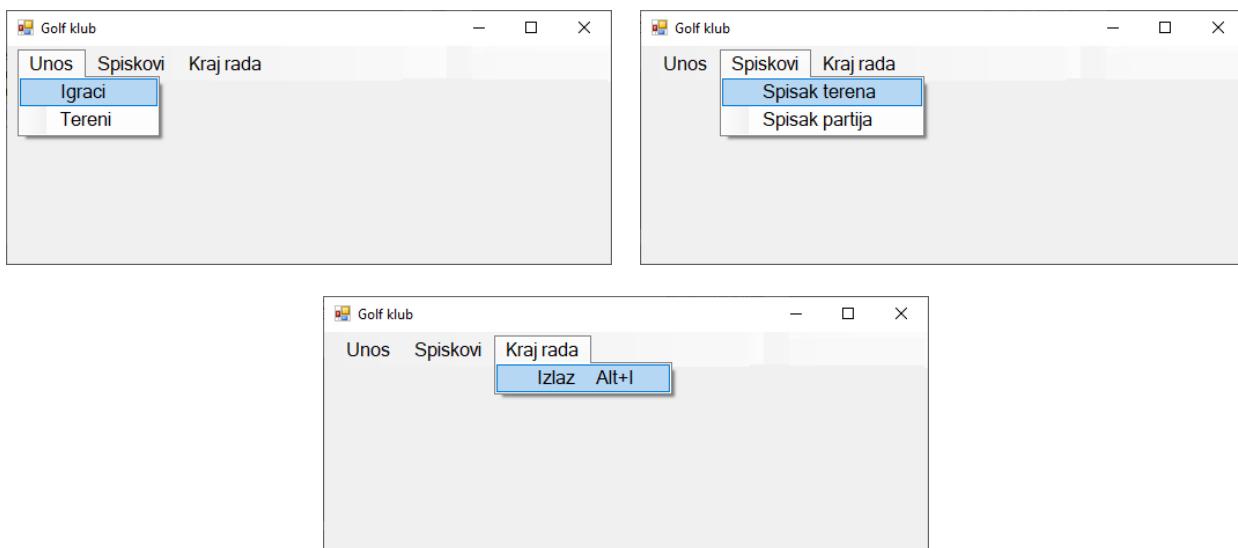
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

У табели партија желимо да чувамо и временске услове у којима је партија одиграна.

За сваки терен желимо да чувамо податке о врсти сваке рупе (на брежуљку, поред језера...), удаљености до сваке рупе и предвиђен број удараца за сваку рупу.

Такође желимо да чувамо у посебној табели резултате које је сваки играч остварио са укупним скором који је остварио за сваку рупу да би смо могли да пратимо напредак наших чланова

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о играчима обавља се преко обрасца чији је изглед дат на слици:

За град се приказују називи градова сортирани у растући низ.

Уколико шифра играча постоји приказују се подаци о њему.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Spisak terena**, приказује списак терена (шифра, назив) и број партија који је одигран на сваком терену у траженом временском интервалу, сортиране по шифри терена у растући низ.

Образац треба да изгледа као на слици:

	TerenID	Teren	Broj
1		Hanover Golf Club	7
2		Abilene Country Club	5
3		Strawberry Valley G...	7
4		Cedar River Country ...	10
5		Boulder Creek Golf ...	7
6		The Wetlands Golf ...	14
7		Grays Harbor Golf C...	5
8		The Bridges Golf Club	7
9		Ruggles Golf Club	7
10		Silver Lake Country	9

	Partija	Datum
78		29.08.2014
487		21.11.2014
193		26.11.2014
50		22.06.2015
339		28.07.2015
309		14.08.2015
276		14.10.2015
465		17.07.2016

Подаци се приказују у контроли за табеларни приказ података.

Ако корисник одабере неку ставку из контроле за табеларни приказ података, у контроли за приказ података у облику листе приказују се шифре партија и датум када су одигране за онај терен који је селектован, сортирани по датуму у растући низ, а затим по партији у растући низ.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б33

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б33
Назив радног задатка	ГОЛФ КЛУБ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за шифру	5	0
Постављена иницијална вредност за град	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 31)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља по уносу шифре	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	8	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Обезбеђено ажурирање података у бази по спецификацији задатка	8	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 7)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за дане у зависности од месеца	3	0

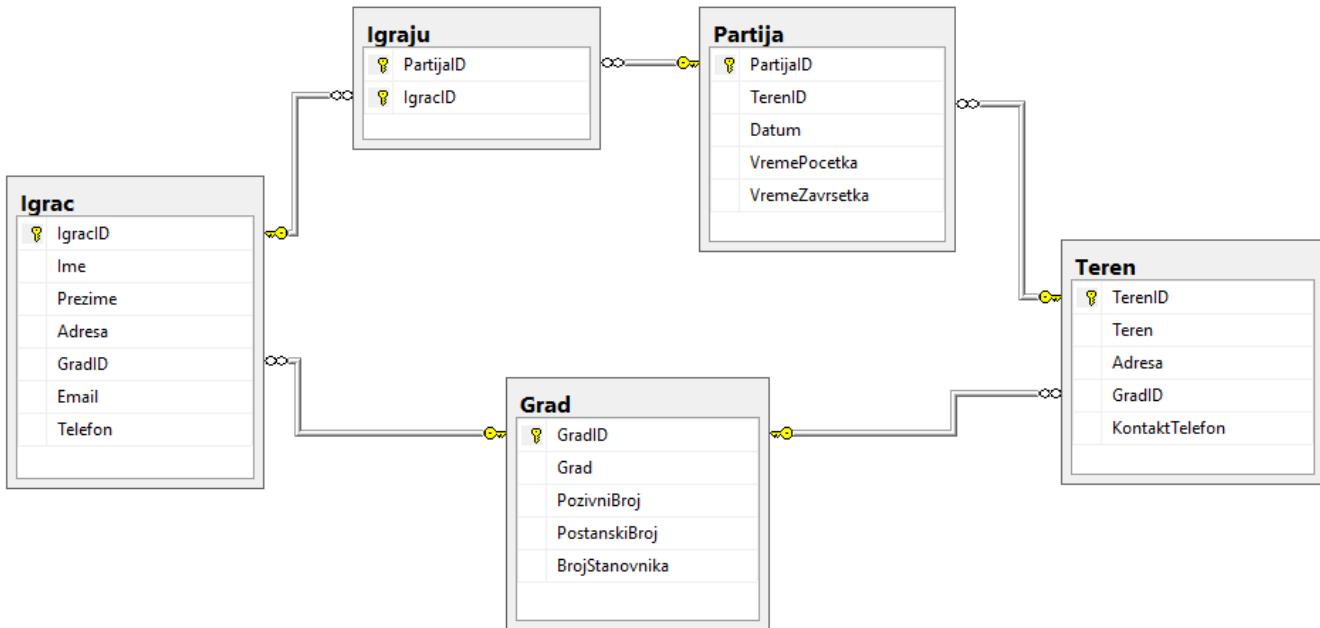
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 27)	ТАЧНО	НЕТАЧНО
Онемогућено уношење погрешног временског интервала	5	0
Приказани тражени подаци у контроли за табеларни приказ података	9	0
Приказани тражени подаци у контроли за приказ података у облику листе	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б34: ГОЛФ КЛУБ

За потребе голф клуба креирана је база података у којој се води евиденција о члановима клуба, теренима и одиграним мечевима.

Дијаграм за постојећу базу дат је на слици:



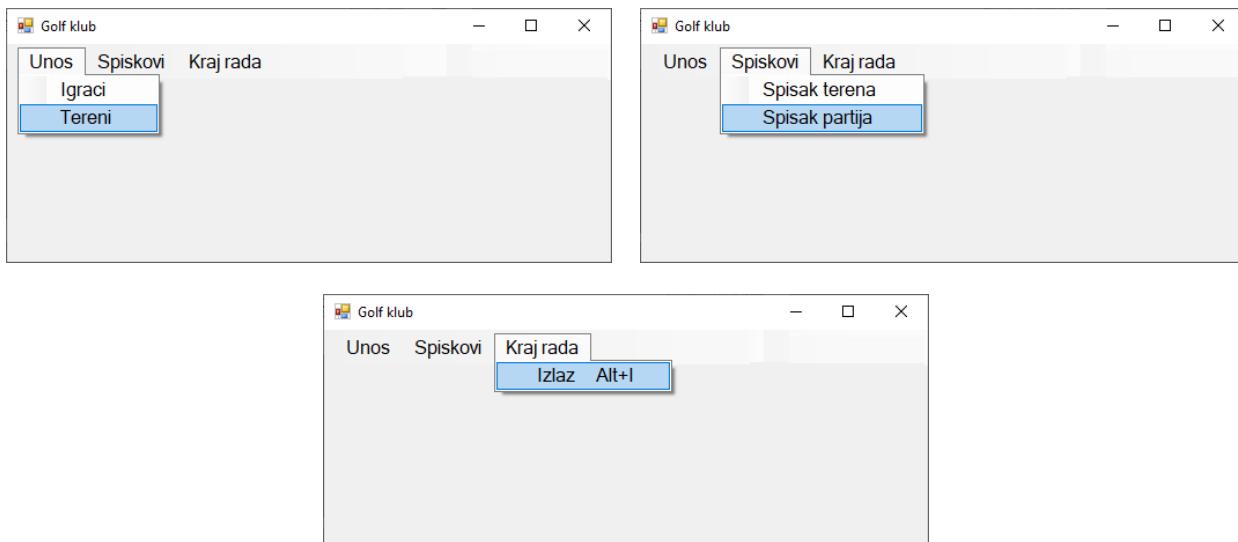
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

У табели партија желимо дачувамо и временске услове у којима је партија одиграна.

За сваки терен желимо да чувамо податке о врсти сваке рупе (на брежуљку, поред језера...), удаљености до сваке рупе и предвиђен број удараца за сваку рупу.

Такође желимо да чувамо у посебној табели резултате које је сваки играч остварио са укупним скором који је остварио за сваку рупу да би смо могли да пратимо напредак наших чланова

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о теренима обавља се преко обрасца чији је изглед дат на слици:

За град се приказују називи градова сортирани у растући низ.

Уколико шифра терена постоји приказују се подаци о њему.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Spisak partija**, приказује листу шифара партија, назив терена на којем је партија одиграна, датум када је одиграна, и укупно време трајања партије, сортиране у опадајући низ по времену трајања партије, по називу терена у растући низ, по шифри партије у растући низ.

Образац треба да изгледа као на слици:

Teren	Datum	Trajanje
High Meadows Golf Club	18.9.2013.	348
The Wetlands Golf Club	22.4.2016.	340
Forest Lake Golf Course	5.6.2017.	330
Glenrochie Country Club	4.10.2015.	320
Maxwell Municipal Golf Course	19.6.2015.	315
Grand Rapids Country Club	8.6.2013.	308
High Meadow Golf Club	26.5.2019.	207

За сате приказати вредности од 1 до 24.

Резултати се приказују у контроли за табеларни приказ података.

За одабрану ставку из контроле за табеларни приказ података приказују се подаци за одабран терен.

Дугме **Izadjii** затвара отворену образац и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б34

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б34
Назив радног задатка	ГОЛФ КЛУБ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за шифру	5	0
Постављена иницијална вредност за град	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 31)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља по уносу шифре	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	8	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Обезбеђено ажурирање података у бази по спецификацији задатка	8	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 7)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за интервале	3	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 27)	ТАЧНО	НЕТАЧНО
Онемогућено уношење погрешног временског интервала	5	0
Приказани тражени подаци у контроли за табеларни приказ података	13	0
Приказани тражени подаци за терен	5	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

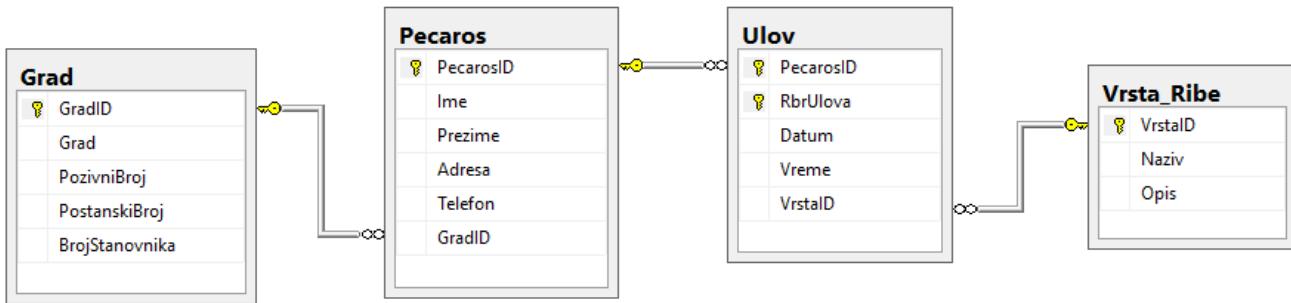
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б35

НАЗИВ РАДНОГ ЗАДАТКА: ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ

ЕИТ-Б35: ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ

За потребе туристичке организације Србије креирана је база података о пецирашима и местима за пецање.

Дијаграм за постојећу базу дат је на слици:

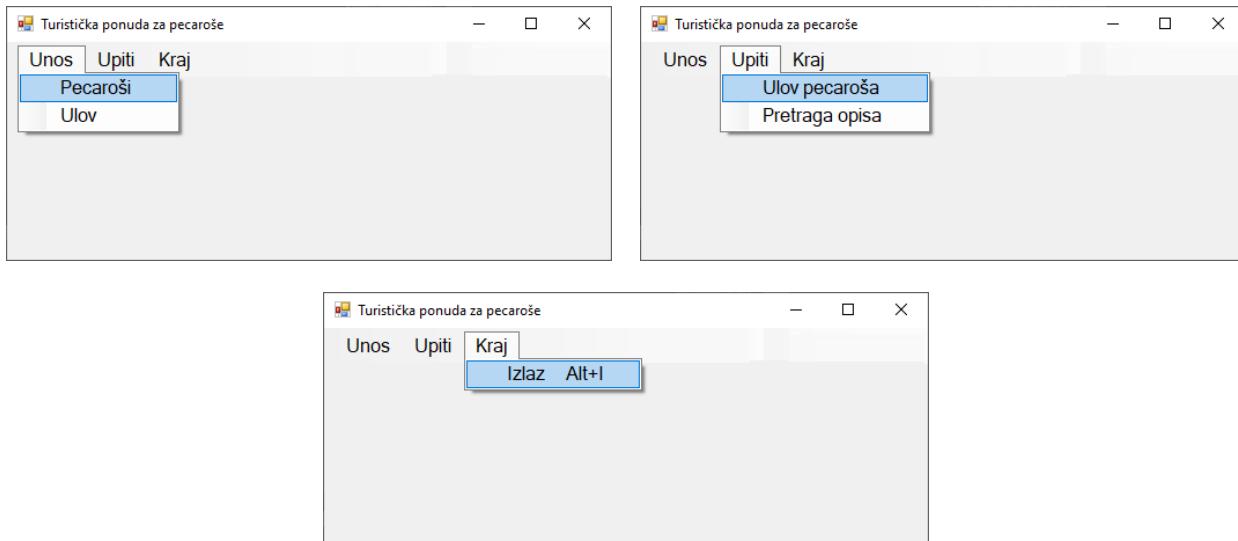


Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

У табели улов желимо да чувамо и тежину рибе коју је пециар је упцео. Тежина се чува у грамима (нпр. 1,250.00).

Желимо да понудимо посетиоцима податке о језерима у околини са бројем локација на сваком језеру на која могу доћи возилом. Потребно је креирати табелу за језеро са називом језера и бројем локација за пецање, као и посебну табелу за сваку локацију где би се чувао назив локације и белешке које су битне за њу (број места за штапове, терен....).

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о пецирашима обавља се преко обрасца чији је изглед дат на слици:

Šifra	Ime	Prezime	Adresa	Grad	Telefon
1	Uroš	Dačić	Cvetna 29	Novi Sad	048-972-554
2	Nikša	Drenovac	Branka Radičevića 72	Prokuplje	071-829-558
3	Kristijan	Beloica	Palih boraca 72	Prokuplje	096-248-217
4	Balša	Danilović	Branka Miljkovića 44	Vrbas	025-314-686
5	Petar	Adamović		Kraljevo	034-212-290
6	Andrej	Filipović	Palih boraca 73	Apatin	027-721-029
7	Mihajlo	Andrić	Somborska 79	Apatin	055-519-457
8	Dušan	Dimitrijević	Lala 44	Vlasotince	015-871-488
9	Lazar	Barjaktarević	Partizanska 41	Negotin	097-676-778

 The last row (Andrej Filipović) is highlighted with a blue background."/>

У оквиру са листом приказују се подаци о пециршима.

За град се приказују називи градова сортирани у растући низ.

Уколико пецирш са унетом шифром постоји, приказују се подаци о њему.

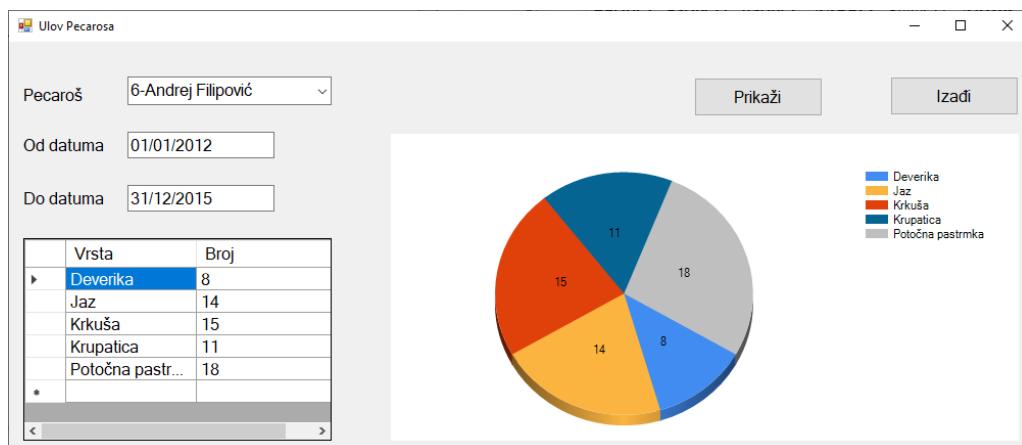
Обезбедити ажурирање оквира са листом.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању/измени података, као и о обрађеним грешкама.

Ставка менија **Ulov pecarosa**, графички приказује укупан број риба који је уловио тражени пецирш у одређеном временском интервалу.

Образац треба да изгледа као на слици:



За пецирша се приказују шифра пецирша и име пецирша сортирани по шифри у растући низ.

Један од датума може бити недефинисан, али не и оба.

Уколико није унет почетни датум приказују се подаци до крајњег датума. Уколико није унет крајњи датум приказују се подаци до данашњег датума.

Резултати упита се приказују у контролиза табеларни приказ података на основу њих црта график.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranom базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б35

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б35				
Назив радног задатка	ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постављена иницијална вредност за град	5	0
Постављена иницијална вредност у оквир са листом	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 33)	ТАЧНО	НЕТАЧНО
Приказани подаци о пеџарошу ако тражена шифра постоји у бази	4	0
Обезбеђено уписивање података у базу по спецификацији задатка	7	0
Додата ставка у оквир са листом после уписа новог слога у базу	2	0
Обезбеђено брисање података из базе по спецификацији задатка	5	0
Обрисана ставка из оквир са листом после брисања слога из базе	2	0
Обезбеђено ажурирање података у бази по спецификацији задатка	7	0
Измењена ставка у оквир са листом после ажурирања слога у бази	2	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за пеџароша	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 23)	ТАЧНО	НЕТАЧНО
Онемогућено уношење погрешног временског интервала	4	0
Приказани тражени подаци у контроли за табеларни приказ података	8	0
Нацртан график са траженим подацима	7	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

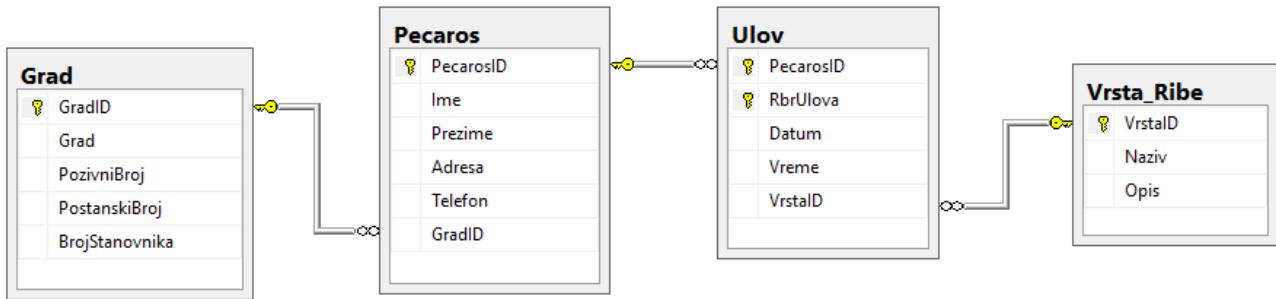
ШИФРА РАДНОГ ЗАДАТКА: ЕИТ-Б36

НАЗИВ РАДНОГ ЗАДАТКА: ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ

ЕИТ-Б36: ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ

За потребе туристичке организације Србије креирана је база података о пецирашима и местима за пецање.

Дијаграм за постојећу базу дат је на слици:

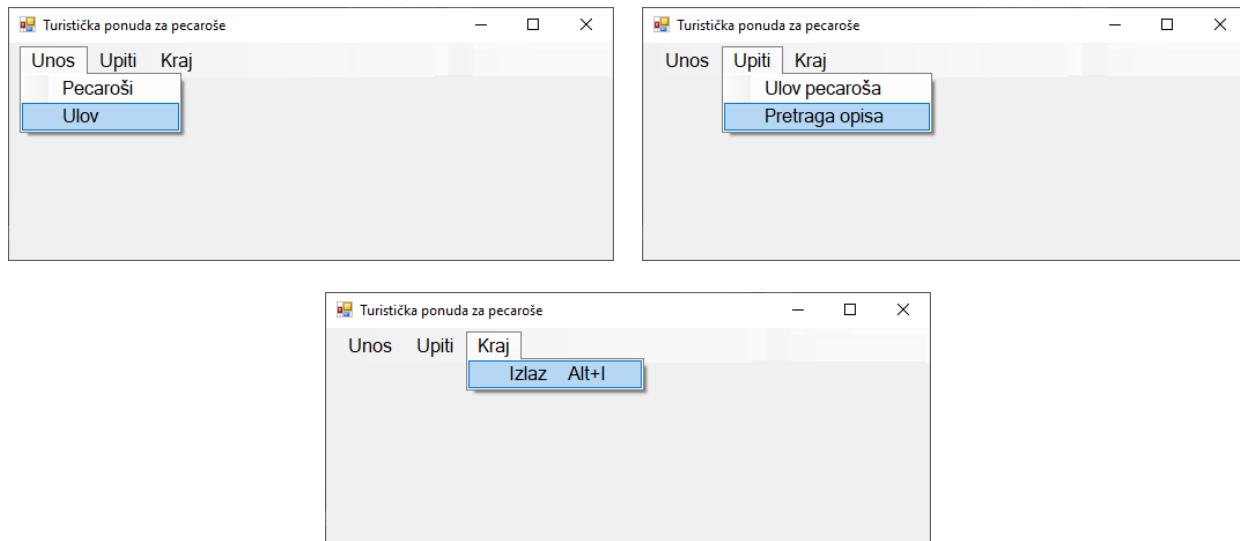


Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

У табели улов желимо да чувамо и тежину рибе коју је пецирош упецао. Тежина се чува у грамима у седећем облику (нпр. 1,250.00).

Желимо да понудимо посетиоцима податке о језерима у околини са бројем локација на сваком језеру на која могу доћи возилом. Потребно је креирати табелу за језеро са називом језера и бројем локација за пецање, као и посебну табелу за сваку локацију где би се чувао назив локације и белешке које су битне за њу (број места за штапове, терен...).

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим сликама:



Уношење података о пецирашима обавља се преко обрасца чији је изглед дат на слици:

The form has two panels:
Left panel:
- **Pecarš**: dropdown menu showing "6-Andrej Filipović".
- **Redni broj ulova**: input field with value "107".
- **Datum**: input field with placeholder "__/__".
- **Vreme**: input field with placeholder "___".
- **Vrsta ribe**: dropdown menu with placeholder "Izaberi vrstu ribe...".
- **Akcija**: radio buttons for "Upiši" (selected) and "Obriši".
- Buttons: **Izvrši** and **Izdaci**.
Right panel:
- **Pecarš**: dropdown menu showing "6-Andrej Filipović".
- **Redni broj ulova**: input field with value "106".
- **Datum**: input field with value "14/02/2015".
- **Vreme**: input field with value "04:00".
- **Vrsta ribe**: dropdown menu showing "Krupatica".
- **Akcija**: radio buttons for "Upiši" (selected) and "Obriši".
- Buttons: **Izvrši** and **Izdaci**.

За пецироша се приказују шифра пецироша, име и презиме пецироша, сортирани по шифри у растући низ.

Заврсту рибе приказју су називи врста риба соритирани у растући низ.

Редни број ставке за сваки улов автоматски се повећава за један код сваког новог уписа.

Могуће је обрисати само последњи улов за изабраног пеџароша.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Pretraga opisa**, омогућава претрагу описа врста риба.

Образац треба да изгледа као на слици:

VrstaID	Naziv	Opis
9	Beli amur	Riba dalekog istoka preneta je u našu ze...
14	Deverika	Deverika je riba šaranskog roda. To je riba ...
12	Jaz	Riba izuzetno ukusnog mesa. Naseljava u...
26	Jezerska pastrmka	Riba brzih i hladnih planinskih vodotoka i l...
10	Krupatice	Vrlo brojna riba. Mnogo liči na deveriku s k...
20	Linjak	Blizak rodak mu je šaran, ali se od njega r...
7	Lipljen	Izrazita sportska riba koju je teško nadmu...
5	Mrena	Sportska riba izduženog vretenastog tela ...
11	Potočna pastrmka	Telo joj je mišićavo sa relativno malom glav...
3	Som	Najveća je slatkvodna evropska riba koja ...
4	Štuka	Štuka je izuzetno zanimljiva riba kako po f...

У контроли за табеларни приказ података приказује се листа шифара, назива и описа оне врсте рибе која задовољава унете критеријуме претраге.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б36

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б36
Назив радног задатка	ТУРИСТИЧКА ПОНУДА ЗА ПЕЦАРОШЕ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије: Место и датум:

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Проширена табела	2	0
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Постаљена иницијална вредност за пецираше	5	0
Постављена иницијална вредност за врсту рибе	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 41)	ТАЧНО	НЕТАЧНО
Приказан редни број ставке за одабраног пецираша уколико је изабрана акција за упис	5	0
Приказан редни број ставке за одабрану резервацију уколико је изабрана акција за брисање	5	0
Омогућено уношење само важећег датума	3	0
Омогућено уношење само важећег времена	3	0
Обезбеђено уписивање података у базу по спецификацији задатка	11	0
Обезбеђено брисање података из базе по спецификацији задатка	10	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 4)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0

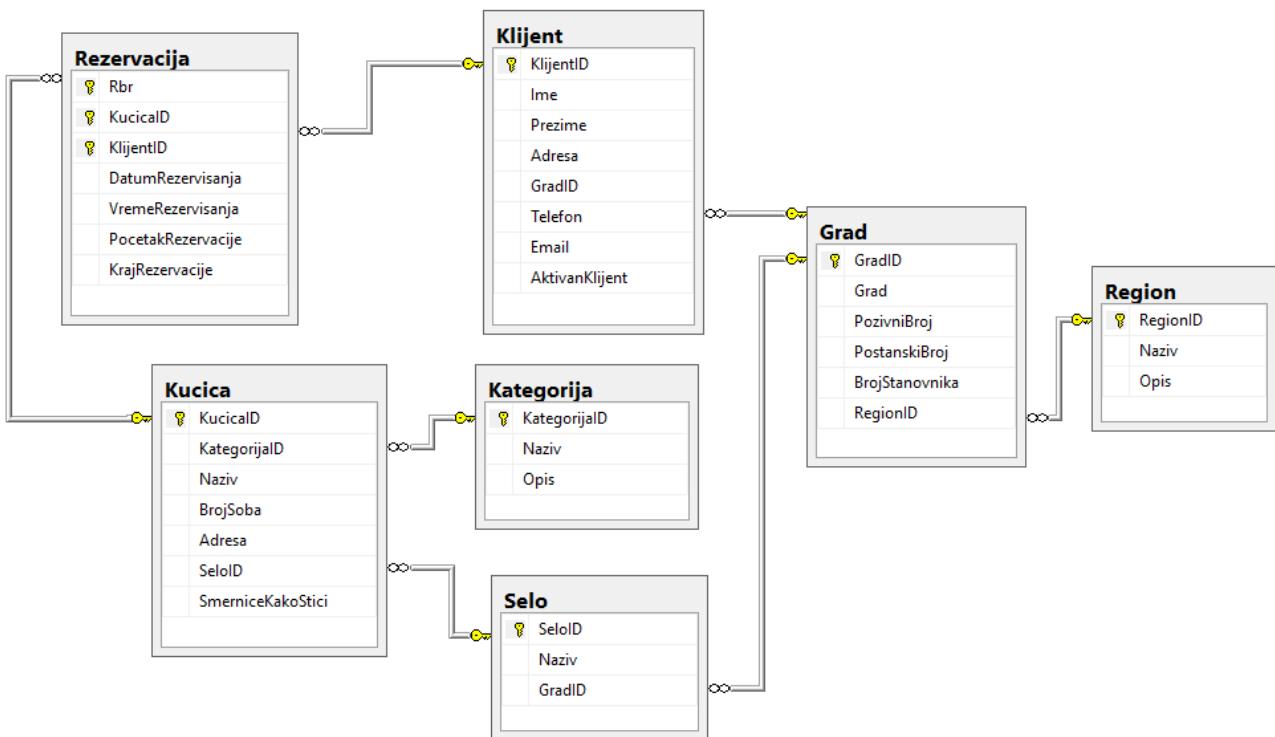
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 20)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци у контроли за табеларни приказ података	16	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б37: СЕОСКИ ТУРИЗАМ

За потребе туристичке организација Србије креирана је база података о сеоском туризму.

Дијаграм за постојећу базу дат је на слици:



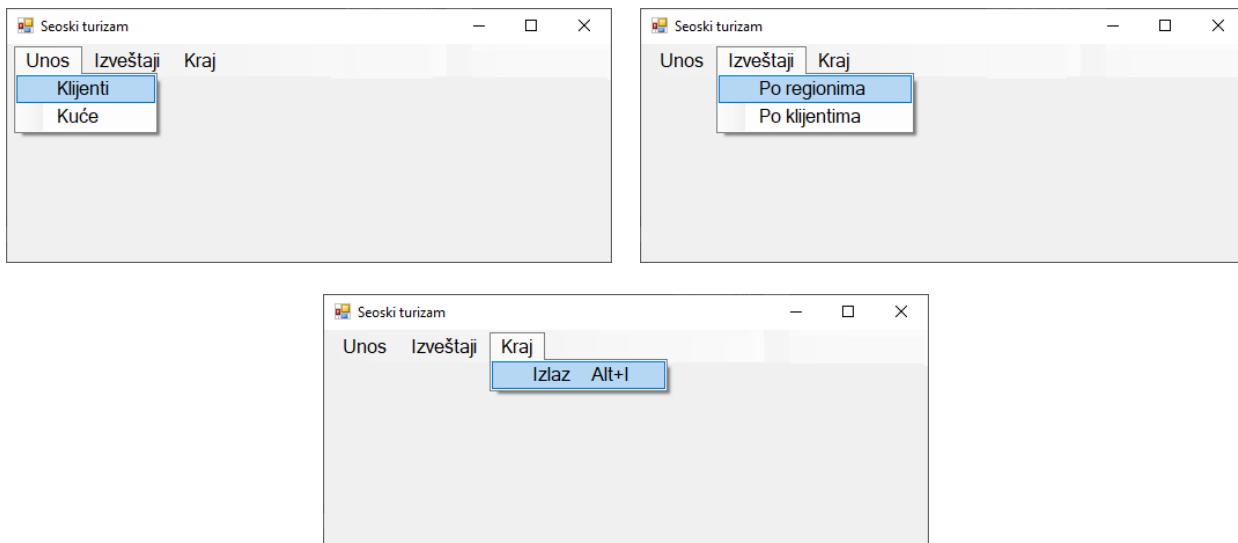
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

За сваку кућицу желимо да чувамо која је цена у којем периоду (лето, зима). Креирати табелу цена која ће имати поља цена, датум од када цена важи и датум до када цена важи.

Такође желимо да клијентима понудимо и списак додатних објеката код сваке кућице (терени за бадмитон, тенис, голф, пешачке стазе, школа јахања, пецање, ски клубови...) са описом објекта.

Обзиром да у оквиру сваке кућице може бити више додатних објеката трансформисати везу више на више увођењем нове табеле кућица додатни објекти у две везе један на више.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о клијентима обавља се преко обрасца чији је изглед дат на слици:

За град се приказују називи градова сортирани у растући низ.

Уколико клијент са унетом шифром постоји, приказати податке о њему.

Притиском на дугме **Nacrtaj** црта се насмејана кућа, на начин који је дат на слици. Обезбедити да дугме **Nacrtaj** може да се притисне само једном.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Po regionima** приказује листу шифара, назива села и укупан број дана који су клијенти провели у сваком од њих, за одабрани регион и у одређеном временском интервалу.

Образац треба да изгледа као на слици:

	Seloid	Naziv	Ukupan broj dana
▶	21	Deliblato	110
	22	Debeljača	220
	69	Rudna glava	121
	87	Skorenovac	79
	97	Zlakusa	182
*			

За регион се приказују називи региона сортирани у растући низ.

Резултати упита се приказују у контролиза табеларни приказ података.

Дугме **Izadjii** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креираном базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б37

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б37
Назив радног задатка	СЕОСКИ ТУРИЗАМ
Назив школе	
Седиште	
Образовни профил	Електротехничар информационих технологија - оглед
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 12)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Дефинисана иницијална вредност за град	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља по уносу шифре	5	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Нацртана слика	7	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за регион	5	0

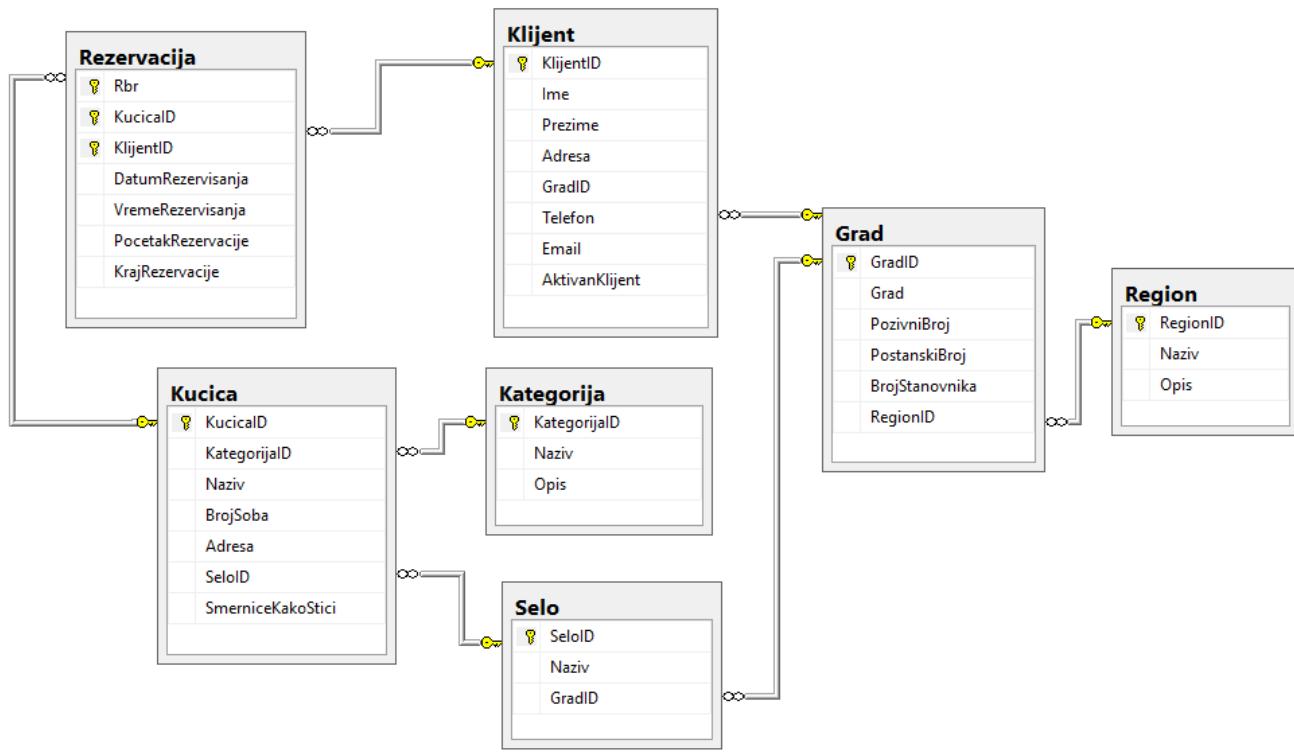
5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 23)	ТАЧНО	НЕТАЧНО
Онемогућено уношење погрешног временског интервала	5	0
Приказани тражени подаци у контроли за табеларни приказ података	14	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

ЕИТ-Б38: СЕОСКИ ТУРИЗАМ

За потребе туристичке организација Србије креирана је база података о сеоском туризму.

Дијаграм за постојећу базу дат је на слици:



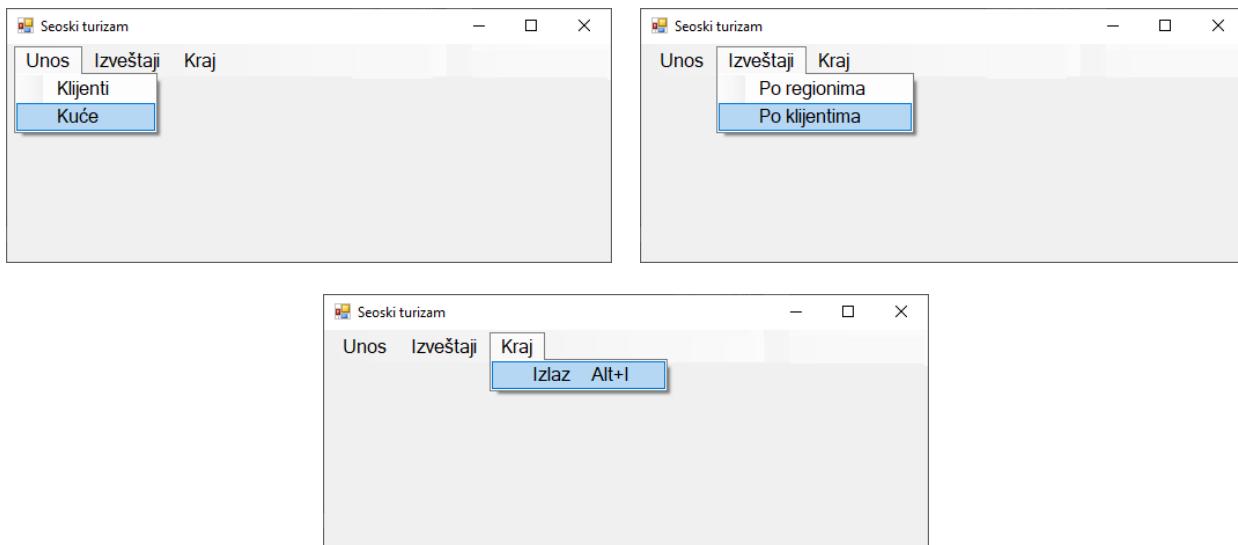
Користећи DDL наредбе проширити базу тако да се задовоље следећи захтеви:

За сваку кућицу желимо да чувамо која је цена у којем периоду (лето, зима). Креирати табелу цена која ће имати поља цена, датум од када цена важи и датум до када цена важи.

Такође желимо да клијентима понудимо и списак додатних објеката код сваке кућице (терени за бадминтон, тенис, голф, пешачке стазе, школа јахања, пецање, ски клубови...) са описом објекта.

Обзиром да у оквиру сваке кућице може бити више додатних објеката трансформисати везу више на више увођењем нове табеле кућица додатни објекти у две везе један на више.

Креирати десктоп апликацију која ће омогућити рад са постојећом базом. Структура менија дата је на следећим slikama:



Уношење података о клијентима обавља се преко обрасца чији је изглед дат на слици:

За категорију се приказују називи категорија сортирани у растући низ.

За село се приказују називи села сортирани у растући низ.

Уколико кућа са унетом шифром шифром постоји, приказати податке о њој.

Притиском на дугме **Nacrtaj** црта се насмејана кућа, на начин који је дат на слици. Обезбедити да дугме **Nacrtaj** може да се притисне само једном.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Обезбедити да корисник добије поруку о успешном упису/брисању података, као и о обрађеним грешкама.

Ставка менија **Po klijentima**, приказује листу назива села и време у коме је одређени клијент из унетог града боравио. Образац треба да изгледа као на слици:

	Naziv	Datum od	Datum do
▶	Bajmok	16.10.2017	26.10.2017
	Perlez	3.8.2015	11.8.2015
	Starčevo	30.11.2016	5.12.2016
*	Bajmok	25.9.2013	7.10.2013

За град се приказују називи градова сортирани у растући низ.

По избору града, за клијента се приказује шифра, име и презиме клијента који долазе из одабраног града.

Резултати упита се приказују у контролиза табеларни приказ података.

Дугме **Izadji** затвара отворен образац, и враћа на почетни мени.

Максимално време за израду задатка је 240 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

Прибор за рад:

Рачунар са потребним програмима, креiranом базом и унетим тест подацима.

За оцењивање користити образац за оцењивање број ЕИТ – Б38

ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА

Шифра радног задатка	ЕИТ-Б38				
Назив радног задатка	СЕОСКИ ТУРИЗАМ				
Назив школе					
Седиште					
Образовни профил	Електротехничар информационих технологија - оглед				
Име и презиме кандидата					
Име и презиме ментора					

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:						Укупно бодова
Аспекти	1.	2.	3.	4.	5.	
Бодови						

Члан испитне комисије:	Место и датум:
------------------------	----------------

КОМЕНТАРИ:

За сваки индикатор заокружити одговарајући број бодова

1. Проширивање базе података

ИНДИКАТОРИ (максималан број бодова 24)	ТАЧНО	НЕТАЧНО
Креирана табела 1	5	0
Дефинисани кључеви у табели 1	3	0
Креирана табела 2	5	0
Дефинисани кључеви у табели 2	3	0
Креирана табела 3	5	0
Дефинисани кључеви у табели 3	3	0

2. Дефинисање иницијалних вредности на делу апликације за унос података

ИНДИКАТОРИ (максималан број бодова 17)	ТАЧНО	НЕТАЧНО
Креиран и оперативан мени	3	0
Креиран образац за унос података са траженим елементима	4	0
Дефинисана иницијална вредност за категорију	5	0
Дефинисана иницијална вредност за село	5	0

3. Креирање дела апликације за унос података

ИНДИКАТОРИ (максималан број бодова 32)	ТАЧНО	НЕТАЧНО
Приказани подаци у одговарајућа поља по уносу шифре	5	0
Обезбеђено уписивање података у базу по спецификацији задатка	9	0
Обезбеђено брисање података из базе по спецификацији задатка	7	0
Нацртана слика	7	0
Обрађена упозорења	2	0
Обрађене грешке	2	0

4. Дефинисање иницијалних вредности на делу апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 9)	ТАЧНО	НЕТАЧНО
Креиран образац са траженим елементима	4	0
Постављена иницијална вредност за град	5	0

5. Креирање дела апликације за упит из базе

ИНДИКАТОРИ (максималан број бодова 18)	ТАЧНО	НЕТАЧНО
Приказани тражени подаци за клијента	5	0
Приказани тражени подаци у контроли за табеларни приказ података	9	0
Обрађена упозорења	2	0
Обрађене грешке	2	0