

LINK ZA ZADATKE



Завод за унапређивање
образовања и васпитања



Центар за стручно
образовање и образовање
одраслих



ZUJA

МАТУРСКИ ИСПИТ
ЕЛЕКТРОТЕХНИЧАР ИНФОРМАЦИОНИХ ТЕХНОЛОГИЈА

Приручник о полагању матурског испита у образовном профилу
Електротехничар информационих технологија

Web Programiranje

Web Dizajn

Београд, јануар 2022.

Садржај:

УВОД.....	1
КОНЦЕПТ МАТУРСКОГ ИСПИТА	2
ЦИЉ	3
СТРУКТУРА.....	3
ОЦЕЊИВАЊЕ СТРУЧНИХ КОМПЕТЕНЦИЈА	3
ПРЕДУСЛОВИ ЗА ПОЛАГАЊЕ И УСЛОВИ СПРОВОЂЕЊА	7
ОРГАНИЗАЦИЈА	7
ЕВИДЕНТИРАЊЕ УСПЕХА И ИЗВЕШТАВАЊЕ	8
ДИПЛОМА И УВЕРЕЊЕ	8
II ИСПИТИ У ОКВИРУ МАТУРСКОГ ИСПИТА	9
1. ИСПИТ ИЗ МАТЕРЊЕГ ЈЕЗИКА И КЊИЖЕВНОСТИ.....	9
2. ИСПИТ ЗА ПРОВЕРУ СТРУЧНО–ТЕОРИЈСКИХ ЗНАЊА	9
3. МАТУРСКИ ПРАКТИЧНИ РАД	11
АНЕКС 1. Стандард квалификације Електротехничар информационих технологија	13
<u>АНЕКС 2. Збирка теоријских задатака</u>	<u>20</u>
АНЕКС 3. Радни задаци са обрасцима за оцењивање	103

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 као и веће 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.

На тим основама је након фазе реализације и евалуације огледа, у подручју рада Електротехника школске 2017/18. године, уведен нови наставни програм⁴: **Електротехничар информационих технологија**. Овај програм развијен је на основу **стандарда квалификације** (датом у **Анексу 1**). Примена стандардизације у систему стручног образовања подразумева и увођење **матурског испита**⁵, којом се обезбеђује провера стечености стручних компетенција прописаних стандардом квалификације.

Прва генерација ученика образовног профила Електротехничар информационих технологија – оглед, завршила је своје школовање полагањем матурског испита школске 2015/16. године.

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

Приручник за полагање матурског испита који је пред Вама је јавни документ намењен ученицима и наставницима средњих стручних школа у којима се спроводи наставни програм Електротехничар информационих технологија, 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 система квалификација, 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 "Сл. Гласник РС – Просветни гласник бр. 1/2018, 9/2020"

⁵ Закон о средњем образовању и васпитању "Службени гласник РС бр.55/2013

КОНЦЕПТ МАТУРСКОГ ИСПИТА

Матурски испит је један од елемената система обезбеђивања квалитета стручног образовања. Полагањем матурског испита у средњем стручном образовању, појединац стиче **квалификацију** неопходну за учешће на тржишту рада.

Матурским испитом се проверава да ли је ученик, по успешно завршеном четврогодишњем образовању, стекао стандардом квалификације прописана знања, вештине, ставове и способности, тј. стручне компетенције за занимање(а) за које се школовао у оквиру образовног профила. Матурски испит састоји се од три независна испита:

- испит из српског језика и књижевности, односно језика и књижевности на којем се ученик школовао (у даљем тексту: матерњи језик);
- испит за проверу стручно–теоријских знања;
- матурски практични рад.

Поред дипломе, сваки појединац полагањем оваквог испита стиче и тзв. додатак дипломи – *Уверење о положеним испитима у оквиру савладаног програма за образовни профил*, чиме се на транспарентан начин послодавцима представљају стечене компетенције и постигнућа ученика.

Концепт матурског испита заснован је на следећим **принципима**:

- уједначавање квалитета матурског испита на националном нивоу,
- унапређивање квалитета процеса оцењивања.

Уједначавање квалитета матурског испита на националном нивоу подразумева спровођење испита по једнаким захтевима и под једнаким условима у свим школама. Увођење механизама осигурања квалитета дефинисаних кроз стандардизоване процедуре и упутства за реализацију, важан су аспект квалитетног спровођења испита. На тај начин се доприноси уједначавању квалитета образовања на националном нивоу за сваки образовни профил.

Унапређивање квалитета процеса оцењивања постиже се применом **методологије оцењивања заснованог на компетенцијама**⁶, као валидног и објективног приступа вредновању компетенција. Развој објективних критеријума процене и одговарајућих метода и инструмената омогућен је успоставом система стандарда квалификације. У складу са тим, оцењивање засновано на компетенцијама почива на операционализацији радних задатака проистеклих из реалних захтева посла односно процеса рада.

Квалитет оцењивања, посебно у домену поузданости и објективности, остварује се и увођењем делимично екстерног оцењивања. Представници послодаваца, стручњаци у одређеној области, обучавају се и учествују као екстерни чланови комисија у оцењивању на матурском испиту.

Резултати матурског испита користе се 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 о полагању матурског испита** (у даљем тексту: Приручник), којим се детаљно описује начин припреме, организације и реализације испита. У састав Приручника улазе: Стандард квалификације Електротехничар информационих технологија, збирка теоријских задатака за матурски испит, листа радних задатака, радни задаци, прилози и обрасци за оцењивање радних задатака.

Приручнике припрема, у сарадњи са тимовима наставника сваког профила, Завод за унапређивање образовања и васпитања – Центар за стручно образовање и образовање одраслих (у даљем тексту: Центар).

⁶ За потребе примене концепта оцењивања заснованог на компетенцијама у стручном образовању и посебно у области испита развијен је приручник „Оцењивање засновано на компетенцијама у стручном образовању“ у оквиру кога су описане карактеристике концепта, његове предности у односу на остале приступе оцењивању, методе примерене таквој врсти оцењивања, као и стандардизован методолошки пут за развој критеријума процене компетенција за одређену квалификацију (www.zuov.gov.rs)

I ПРОГРАМ МАТУРСКОГ ИСПИТА

ЦИЉ

Матурским испитом проверава се да ли је ученик, по успешно завршеном образовању за образовни профил Електротехничар информационих технологија, стекао стручне компетенције прописане Стандардом квалификације Електротехничар информационих технологија⁷.

СТРУКТУРА

Матурски испит састоји се од три независна испита:

- испит из матерњег језика и књижевности;
- испит за проверу стручно–теоријских знања;
- матурски практични рад.

ОЦЕЊИВАЊЕ СТРУЧНИХ КОМПЕТЕНЦИЈА

У оквиру матурског испита се проверава стеченост **стручних компетенција**. Оцењивање стручних компетенција врши се комбинацијом метода: тестирање стручно теоријских знања и симулација путем извођења практичних радних задатака. Тест знања заснива се на исходима стручног образовања (исходи знања), док су радни задаци формирани превасходно на основу јединица компетенција и омогућавају проверу оспособљености ученика за примену знања, демонстрацију вештина и професионалних ставова у радном контексту. На овај начин је омогућено мерење знања, вештина, ставова и способности који одговарају Стандарду квалификације **Електротехничар информационих технологија**.

Критеријуми оцењивања стручних компетенција развијени су на основу јединица компетенција и чине *Оквир за оцењивање компетенција за квалификацију електротехничар информационих технологија* (у даљем тексту: *Оквир*). Оквир садржи критеријуме процене, дате у две категорије: аспекти и индикатори процене. Инструменти за оцењивање стручних компетенција – обрасци који се користе на матурском испиту формирани су и усклађени са Оквиром.

⁷Стандард квалификације Електротехничар информационих технологија дат је у Анексу 1 овог Приручника

Оквир за оцењивање компетенција за квалификацију Електротехничар информационих технологија ⁸

Табела: за дужност/компетенцију: Припрема и организација рада

Аспекти	индикатори				
	1	2	3	4	5
1. Аспект Припремање и одржавање радног места	Извршен избор и подешавања радног окружења	Преузети потребни ресурси	Постављена резервна копија базе података		
2. Аспект Комуницирање са сарадницима и клијентима	Прикупљени и анализирани захтеви корисника	Презентовање идејног решења (тј. плана активности)	Демонстрирање функционалности реализованог програмског решења		

Табела: за дужност/компетенцију: Израда десктоп апликација

Аспекти	индикатори				
	1	2	3	4	5
1. Аспект Израђивање решења дизајна десктоп апликације	Креирање форме апликације са потребним контролама	Постављени иницијални подаци у контроле на форми /формама	Подешен приказ података уз поштовање принципа прегледности и уредности форме	Извршена заштита апликације на нивоу форме	
2. Аспект Повезивање десктоп апликације са базом података	Приказани подаци добијени упитом над једном или више табела у бази података	Остварена функционалност уписа, измене и/или брисања података у бази података	Извршено освежавање приказа на форми у складу са променом у бази		
3. Аспект Тестирање функционалности десктоп апликације	Обезбеђена валидација унетих података	Реализовано приказивање упозорења у случају настале грешке	Извршен одабир примера за тестирање којима демонстрира заштиту апликације од грешака		

⁸За потребе реализације матурског испита и процену компетентности ученика кроз одговарајуће радне задатаке, извршено је обједињавање компетенција из Стандарда квалификације м и дефинисани су одговарајући аспекти и индикатори.

Аспекти	индикатори				
	1	2	3	4	5
1. Аспект Израђивање дизајнерског решења веб апликације	Креирање веб странице према захтеву	Постављен одговарајући садржај на странице	Извршено стиловање постављеног садржаја према опису у захтеву	Обезбеђено функционисање свих веза између страница	
2. Аспект Програмирање функционалности веб апликације	Рализована тражена функционалност приказивања и/или измена садржаја извора података	Постављени иницијални подаци у контроле веб странице	Реализовано исправно функционисање свих контрола на веб страници према захтеву	Обезбеђено правилно функционисање свих линкова у апликацији	Веб апликација постављена на веб сервер
3. Аспект Организовање података веб апликације	Ресурси веб апликације (слике, фајлови...) су смештени у посебне фолдере	CSS код смештен у посебном фајлу	Код за функционалност и изглед организован у одговарајућим датотекама		

Аспекти	индикатори				
	1	2	3	4	5
1. Аспект Модификовање структуре базе података у складу са захтевима корисника	Анализира захтев за проширење базе и црта изменењен ЕР дијаграм базе	Креира нову табелу и повезује је са постојећим табелама у бази	Додаје нову колону постојећој табели у бази	Додаје ограничење на табелом у бази	
2. Аспект Манипулација подацима у бази података	Креирани прости упити за претрагу података по задатим критеријумима	Креирани „cross“ упити за селекцију података над више табела	Креирани командни упити за упис, измену и/или брисање података из табеле	Тестира функционалност упита	

Табела: за дужност/компетенцију: Израда техничке документације

Аспекти	индикатори				
	1	2	3	4	5
1. Аспект Израђивање делова проектне документације	Користи одговарајуће софтверске алате за израду техничке документације	Израђено техничко упутство за коришћење	Постављено корисничко упутство на посебну форму / веб страницу		

ПРЕДУСЛОВИ ЗА ПОЛАГАЊЕ И УСЛОВИ СПРОВОЂЕЊА

Ученик може да полаже матурски испит у складу са Законом.

Предуслови за полагање и услови за спровођење матурског испита дати су у следећој табели.

Ученик:	
општи:	успешно завршен четврти разред образовног профиле Електротехничар информационих технологија
посебни:	прибор за писање (обавезна хемијска оловка)
Школа:	
за припрему и спровођење матурског испита неопходно је да школа, у договору са социјалним партнерима, обезбеди потребне услове за израду одговарајућих радних задатака:	
<ul style="list-style-type: none">• време (термине за извођење свих делова матурског испита, укључујући план реализације радних задатака);• одговарајући број примерака тестова;• просторе за реализацију теста знања и радна места за реализацију практичног дела 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;• описе радних задатака 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.

Током реализације матурског испита није дозвољена употреба мобилних телефона.

ОРГАНИЗАЦИЈА

Организација матурског испита спроводи се у складу са *Правилником о програму матурског испита за образовни профил Електротехничар информационих технологија*. Матурски испит се организује у школама у три испитна рока који се реализују у јуну, августу и јануару.

Школа благовремено планира и припрема људске и техничке ресурсе за реализацију испита и израђује распоред полагања свих испита у оквиру матурског испита.

За сваку школску годину директор, на предлог наставничког већа, формира Испитни одбор. Испитни одбор чине чланови свих испитних комисија, а председник Испитног одбора је по правилу директор школе.

За сваког ученика директор школе именује **ментора / менторе**. Ментор је наставник стручних предмета који је обучавао ученика у току школовања. Он помаже ученику у припремама за полагање теста за проверу стручно-теоријских знања и матурског практичног рада. У оквиру три недеље планиране наставним планом за припрему и полагање матурског испита, школа организује консултације, информиште кандидате о критеријумима оцењивања и обезбеђује услове (време, простор, опрема) за припрему ученика за све задатке предвиђене матурским испитом.

У периоду припреме школа организује обуку чланова комисија за оцењивање на матурском испиту уз подршку стручних сарадника школе.

Матурски испит спроводи се у школи и просторима где се налазе радна места и услови за реализацију матурског практичног рада.

Матурски испит за ученика може да траје највише четири дана. У истом дану ученик може да полаже само један од делова матурског испита.

За сваки део матурског испита директор школе именује стручну испитну комисију, коју чине три члана и три заменика. Ради ефикасније реализације матурског испита, ако за то постоје прописани кадровски и материјални услови, у школи се може формирати и више испитних комисија, које могу истовремено и независно да обављају оцењивање.

ЕВИДЕНТИРАЊЕ УСПЕХА И ИЗВЕШТАВАЊЕ

Ученик који испуњава ошти услов за приступање матурском испиту дужан је да школи поднесе писану пријаву за полагање и пратећу документацију у складу са Законом. Рок за пријављивање испита одређује школа.

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

- писани састав из материјег језика;
- оцењен тест са испита за проверу стручно - теоријских знања;
- обрасци за оцењивање сваког појединачног радног задатака свих чланова комисије.

Након реализације појединачног испита у саставу матурског испита комисија утврђује и евидентира успех ученика у Записницима о полагању матурског испита и ти резултати се објављују, као незванични, на огласној табли школе.

На основу резултата свих појединачних испита Испитни одбор утврђује ошти успех ученика на матурском испиту. Након седнице испитног одбора на којој се разматра успех ученика на матурском испиту, на огласној табли школе објављују се званични резултати ученика на матурском испиту.

Ошти успех на матурском испиту исказује се једном оценом као аритметичка средња вредност оцена добијених на појединачним испитима у саставу матурског испита.

Ученик је положио матурски испит ако је из свих појединачних испита у саставу матурског испита добио позитивну оцену.

Ученик који је на једном или два појединачна испита у саставу матурског испита добио недовољну оцену упућује се на полагање поправног или поправних испита у саставу матурског испита.

У року од 24 сата од објављивања званичних резултата ученик има право подношења жалбе директору школе на добијену оцену на матурском испиту.

Након реализације испита, а на захтев Центра, школа је у обавези да резултате испита достави Центру, ради праћења и анализе матурског испита. У ту сврху Центар благовремено прослеђује школи одговарајуће обрасце и инструменте за праћење.

ДИПЛОМА И УВЕРЕЊЕ

Ученику који је положио матурски испит издаје се *Диплома о стеченом средњем образовању за образовни профил Електротехничар информационих технологија*.

Уз Диплому школа ученику издаје *Уверење о положеним испитима у оквиру савладаног програма за образовни профил Електротехничар информационих технологија*.

II ИСПИТИ У ОКВИРУ МАТУРСКОГ ИСПИТА

1. ИСПИТ ИЗ МАТЕРЊЕГ ЈЕЗИКА И КЊИЖЕВНОСТИ

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

СТРУКТУРА ИСПИТА

Испит из матерњег језика полаже се писмено.

На испиту ученик обрађује 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.

ОЦЕЊИВАЊЕ

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

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

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 одабраних 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 изведенih јединствених оцена за сваког од ученика председник испитне комисије сумира резултате и предаје потписане записнике и ученичке радове председнику Испитног одбора.

2. ИСПИТ ЗА ПРОВЕРУ СТРУЧНО–ТЕОРИЈСКИХ ЗНАЊА

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

СТРУКТУРА ИСПИТА

У наставном плану и програму за образовни профил **Електротехничар информационих технологија**, стручна знања неопходна за обављање послова и задатака **Електротехничар информационих технологија** стичу се у оквиру предмета: **Програмирање, Веб дизајн, Веб програмирање, Базе података**.

Провера остварености очекиваних исхода знања, односно стручно–теоријских знања врши се завршним тестирањем. Тест садржи највише 50 задатака, а конципиран је тако да обухвата све нивое знања и све садржаје који су процењени као темељни и од суштинског значаја за обављање послова и задатака у оквиру датог занимања, као и за наставак школовања у матичној области.

Тест и кључ за оцењивање теста припрема Центар, на основу Збирке теоријских задатака за матурски испит (Анекс 2) и доставља га школама. Комбинација задатака за матурски тест, узимајући у обзир и критеријум сазнајне сложености, формира се од: познатих задатака из Збирке теоријских задатака за матурски испит (75 бодова) и делимично познатих задатака насталих делимичном изменом задатака из Збирке теоријских задатака за матурски испит (25 бодова). Збирку су, уз координацију Центра, припремили наставници школа у којима се реализује образовни програм за Електротехничар

информационих технологија.

ОЦЕЊИВАЊЕ

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

Укупан број бодова на тесту који ученик може да постигне је **100** и једнак је збире 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 следећој табели.

Укупан број бодова остварен на тесту	УСПЕХ
до 50	недовољан (1)
50,5 – 63	довољан (2)
63,5 – 75	добр (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.

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, школе које имају кандидате у датом испитном року заједнички утврђују и достављају га Центру најкасније седам 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, дан раније у односу на утврђен датум за полагање теста, а кључ на дан реализације теста.
- Лице задужено за техничку припрему теста у школи обавља све припреме и умножава тест. Припремљени тестови се пакују у коверат који се затвара, печати и чува 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 и самостално решава тест. 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 тестирања дежурни наставници записнички предају директору или другом одговорном лицу све решаване и неискоришћене тестове. На огласној табли школе, објављује се кључ теста.
- Председник комисије за преглед тестова преузима Записнике о полагању матурског испита, као и коверат са решаваним тестовима, као и коверат са три примерка кључа (за сваког члана) и комисија приступа прегледу тестова. Након завршеног прегледања, евидентирања и потписивања Записника о полагању матурског испита, формира се извештај о резултатима ученика и постигнутом успеху на испиту за проверу стручно-теоријских знања и достављају потписани записници и сви решавани тестови председнику Испитног одбора.
- Најкасније у року од 24 сата по завршетку реализације теста објављују се незванични резултати тестирања на огласној табли школе.

3. МАТУРСКИ ПРАКТИЧНИ РАД

Циљ матурског практичног рада је провера стручних компетенција прописаних Стандардом квалификације Електротехничар информационих технологија.

СТРУКТУРА ИСПИТА

На матурском практичном раду ученик извршава два комплексна радна задатка којима се проверава стеченост свих прописаних стручних компетенција. Радни задаци се реализују кроз практичан рад.

За проверу прописаних компетенција, на основу Оквира за процену компетенција за квалификацију Електротехничар информационих технологија утврђује се листа комбинација радних задатака.

Листу комбинација радних задатака за проверу компетенција, радне задатке, и инструменте за оцењивање радних задатака припрема Центар у сарадњи са тимовима наставника.

Листа радних задатака и комбинације дате су у Анексу 3 овог Приручника. Остало потребна документација за реализацију практичних задатака налази се у Анексу 4.

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, од којих је један председник комисије
- представник послодавца – компетентни извршилац датих послова у области електротехнике и рачунарства кога предлаже Унија послодаваца Србије у сарадњи са одговарајућим пословним удружењима, Привредном комором Србије и Центром⁹.

Сваки члан испитне комисије пре испита добија обрасце за оцењивање радних задатака у оквиру одабране комбинације, а председник комисије води одговарајући део Записника о полагању матурског испита.

Сваки члан комисије индивидуално оцењује рад ученика, користећи одговарајући образац за оцењивање радног задатка¹⁰.

Сваки радни задатак може се оценити са **највише 100 бодова**. Сваки члан испитне комисије вреднујући појединачно индикаторе у свом обрасцу за оцењивање радног задатка утврђује укупан број бодова који је ученик остварио у оквиру појединачног задатка.

Појединачан број бодова (сваког члана комисије) се уноси на одговарајуће место у Записнику о полагању матурског испита и на основу тога комисија утврђује просечан број бодова за сваки радни задатак.

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

Уколико је просечан број бодова који је кандидат остварио на појединачном радном задатку 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 преводи се у успех. Скала успешности је петостепена и приказана је у следећој табели.

УКУПАН БРОЈ БОДОВА	УСПЕХ
0-99	недовољан (1)
100-125	довољан (2)
126-151	добар (3)
152-177	врло добар (4)
178-200	одличан (5)

⁹ Сагласност на чланство представника послодавца у комисији, на предлог школа, даје Унија послодаваца Србије односно Привредна комора Србије у сарадњи са Заводом за унапређивање образовања и васпитања - Центром. Базу података о екстерним члановима испитних комисија води Центар.

¹⁰У оквиру Анекса3 овог Приручника налазе се обрасци за оцењивање радних задатака

ОРГАНИЗАЦИЈА ИСПИТА

- Матурски практичан рад реализује се у школским 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 радне задатке на основу листе задатака из овог Приручника и формира **школску листу** која ће се користити у том испитном року. Број комбинација мора бити најмање за 10% већи од броја ученика који полажу матурски испит у једном одељењу.
- По формирању Испитног одбора директор утврђује чланове комисија за оцењивање матурског практичног рада и њихове заменике. Предлог имена екстерних чланова комисије се благовремено доставља Центру ради добијања сагласности.
- По избору чланова комисије за оцењивање, школа треба да изврши обуку чланова комисије уз подршку стручних сарадника школе. Сви чланови комисије треба да буду упознати са документом *Инструкције за оцењиваче* и да у складу са тим усвоје ток припреме и извођења радних задатака, као и да примене утврђене принципе и правила оцењивања.
- Лице задужено за техничку подршку реализацији матурског практичног рада припрема неозначене коверте у којима се за сваки задатак налазе по четири описа задатака (један ће преузети ученик, а три су намењена члановима комисије) и три обрасца за оцењивање, са претходно унетим подацима о школи, шифром и називом задатка, за чланове комисије. За чланове комисије је могуће припремити описе задатака у електронској форми, уз услов да сваки члан комисије има самосталан приступ рачунару, таблету или лаптопу.
- Непосредно пред полагање ученик извлачи комбинацију радних задатака, без права замене. Додељује му се опис једног радног задатка, а опис другог радног задатка непосредно пред извршење тог задатка. Чланови комисије преузимају опис задатка и листе за оцењивање и у заглављу уносе учениково име.
- Сваком ученику се обезбеђују **једнаки услови** за почетак обављања радног задатка.
- Трочлана комисија прати рад сваког ученика током реализације практичног рада.
- Ученички продукти у процесу израде задатка (датотеке) се чувају у одговарајућој форми у складу са законом.
- Најкасније у року од 24 сата по завршетку реализације практичног матурског рада сумирају се резултати тог дела испита и објављују, као незванични, на огласној табли школе. Потписани записници, са предвиђеном документацијом, прослеђују се председнику Испитног одбора.

**АНЕКС 1. СТАНДАРД КВАЛИФИКАЦИЈЕ ЕЛЕКТРОТЕХНИЧАР
ИНФОРМАЦИОНИХ ТЕХНОЛОГИЈА**

РАДНА ВЕРЗИЈА

СТАНДАРД КВАЛИФИКАЦИЈЕ

ЕЛЕКТРОТЕХНИЧАР ИНФОРМАЦИОНИХ ТЕХНОЛОГИЈА

1. Назив квалификације: Електротехничар информационих технологија

2. Сектор - подручје рада: Електротехника

3. Ниво квалификације: IV

4. Начин стицања квалификације:

Квалификација се стиче након успешног завршеног процеса средњег стручног образовања.

5. Трајање:

Програм средњег стручног образовања за стицање квалификације траје четири године.

6. Начин провере:

Достигнутост исхода програма средњег стручног образовања се проверава на матурском испиту који спроводи средња школа и Завод за унапређивање образовања и васпитања (ЗУОВ).

7. Заснованост квалификације:

Квалификација се заснива на опису рада, циљевима стручног образовања и исходима стручног образовања.

7.1. Опис рада

Дужности - стручне компетенције:

- Припрема и организација рада
- Израда десктоп апликација
- Израда веб апликација (статичких и динамичких веб страница)
- Израда и управљање базом података
- Израда техничке документације

Дужности - стручне компетенције	Задаци - јединице компетенција
Припрема и организација рада	<ul style="list-style-type: none"> – Припрема и одржава потребну опрему, уређаје и софтвер – Прикупља захтеве клијената, корисника и менаџера пројекта и анализира их у оквиру тима – Учествује у тиму и ефикасно комуницира са сарадницима – Комуницира и уважава захтеве клијента – Спроводи планиране задатке у складу са стандардима квалитета – Прати иновације у области информационих система и предлаже их за увођење у радни процес
Израда десктоп апликација	<ul style="list-style-type: none"> – Израђује предлог решења десктоп апликације – Израђује софтверске модуле за апликацију – Повезује апликацију са базом података – Спроводи поступке за заштиту од губитака и неовлашћеног приступа корисничких подацима – Тестира и врши процену ефикасности примене апликације (време, системски ресурси, енергија...) – Спроводи корекције и поправке у апликацији – Инсталарира и конфигурише решење/апликацију
Израда веб апликација (статичких и динамичких веб страница)	<ul style="list-style-type: none"> – Израђује предлог решења веб апликације – Израђује делове веб апликације – Повезује делове веб апликације у функционалну целину у складу са захтевом клијента и дизајнера – Врши подешавања и оптимизацију веб апликације – Спроводи сигурносне механизме за заштиту података на вебу – Повезује веб апликацију са базом података – Тестира решење и евалуира резултате – Инсталарира решење на веб сервер
Израда и управљање базом података	<ul style="list-style-type: none"> – Израђује структуру једноставне базе података – Манипулише садржајима у бази података – Врши измену структуре базе података – Одржава функционисање базе података (оптимизација, поправка...) – Врши сигурносну администрацију база података (права приступа, сигурносне копије, враћање података)
Израда техничке документације	<ul style="list-style-type: none"> – Учествује у припреми пројектне документације – Припрема елементе за обрачун цене израде решења – Израђује техничка упутства за коришћење – Чува и архивира техничку документацију – Води евиденције у складу са процедурима организације (извршени задаци, радни налози, записник...)

7.1.1. Изложеност ризицима при обављању дужности:

- ризик од стреса
- ризик од оштећења вида

7.2. Циљеви стручног образовања

Циљ стручног образовања за квалификацију **ЕЛЕКТРОТЕХНИЧАР ИНФОРМАЦИОНИХ ТЕХНОЛОГИЈА** је оспособљавање лица за израду десктоп и веб апликација и израду и управљање базом података.

Неопходност сталног прилагођавања променљивим захтевима тржишта, потребе континуираног образовања, стручног усавршавања, развоја каријере и сл. усмерава да лица буду оспособљавана за:

- примену теоријских знања у практичном контексту;
- аналитичко размишљање и решавање проблема;
- тимски рад;
- примену мера заштите здравља и животне средине у процесу рада;
- преузимање одговорности за властито континуирано учење и напредовање у послу и каријери;
- препознавање пословних могућности у радној средини и ширем социјалном окружењу.

7.3. Исходи стручног образовања

Стручне компетенције	Знања	Вештине	Способности и ставови
По завршеном програму образовања, лице ће бити у стању да:			
Припрема и организација рада	<ul style="list-style-type: none">- објасни намену и функције компоненти рачунарског система;- наведе делове системског софтвера и објасни њихове функције- познаје архитектуру оперативних система- наведе и објасни улогу интерфејса рачунара- разликује, објасни и користи различите начине приступа Интернету;- објасни задатак IP протокола и појам рутирања;	<ul style="list-style-type: none">- обавља инсталацију update и upgrade софтвера у циљу отклањања проблема у раду или у циљу побољшања перформанси рада;- конфигурише основне поставке оперативног система- конфигурише мрежне уређаје- врши избор уређаја у зависности од захтева- конфигурише веб сервер- управља корисничким правима и налозима- подешава развојно окружење	<ul style="list-style-type: none">- савесно, одговорно, уредно и прецизно обавља поверене послове;- ефикасно планира и организује време;- испољи позитиван однос према значају спровођења важећих стандарда у раду;- испољи позитиван однос према функционалности и техничкој исправности опреме и уређаја

Стручне компетенције	Знања	Вештине	Способности и ставови
	<ul style="list-style-type: none"> - познаје развојно окружење - објасни основе менаџмента услуга / производње - објасни значај информационих технологија за савремено пословање - наведе главне карактеристике квалитета софтвера 	<ul style="list-style-type: none"> - самостално прикупља податке са тржишта - прави понуду услуге; - изради једноставан пословни план - примењује стандарде квалитета приликом израде информационог система - анализира захтеве клијената и предлаже најоптималнија решења - учествује у раду тима, предлаже решења унутар тимског рада и дискутује о њима - користи научене комуникационе вештине у пословној околини - прати и усваја нова технолошка достигнућа у подручју информационих система - користи стручну литературу и претраживање доступних база инфорамција и база знања 	<ul style="list-style-type: none"> - које користи при обављању послана; - испољи љубазност, комуникативност, предузимљивост, флексибилност у односу према сарадницима; - испољи креативност и иновативност при обављању послана; - оријентисан према клијенту и прилагодљив на промене у раду; - решава проблеме у раду; - испољи позитиван однос према професионално-етичким нормама и вредностима; - испољи позитиван однос према мерама безбедности и заштите података.
Израда десктоп апликација	<ul style="list-style-type: none"> - разликује основне алгоритамске структуре - дефинише израз на основу кога се одређује ток извршавања алгоритма и програма - познаје и примењује технике рада са низовима - разуме појам декомпозиције проблема - разуме шта су датотеке и чему служе - дефинише, креира и користи структуру и низове структуре - разуме разлику класе у односу на структуру - разуме и имплементира обраду грешке - разуме смисао наслеђивања класа - разуме употребу апстрактних класа и интерфејса - разуме смисао програмирања вођеног догађајима 	<ul style="list-style-type: none"> - комбинује различите алгоритамске структуре - користи различите типове гранања - користи наредбе за формирање и излаз из бројачких циклуса - користи показивачке променљиве - креира функције при чему разуме и користи пренос параметара по вредности и по референци - самостално отвара и креира датотеке - изводи операције претраживања датотеке - пише и тестира програме у којима се демонстрира примена класе - пише и тестира програме у којима се користи руковање изузецима - пише и тестира програме у којима се користи наслеђивање - креира апликацију уз примену стандардних компоненти из библиотеке компоненти 	

Стручне компетенције	Знања	Вештине	Способности и ставови
	<ul style="list-style-type: none"> - примењује динамичко креирање компоненти - разликује конекциони и бесконекциони режим приступа подацима у бази - познаје и користи класе ADO.NET архитектуре - планира и израђује интерфејс за комуникацију са базом 	<ul style="list-style-type: none"> - врши читање и анализу прочитаних података из базе - врши упис, ажурирање и брисање података у бази 	
Израда веб апликација (статичких и динамичких веб страница)	<ul style="list-style-type: none"> - разуме разлику између статичких и динамичких сајтова - разуме функцију веб сервера - познаје структуру и користи системе за управљање садржајима (CMS) - налази најбоље решење за структуру веб презентације у зависности од потребе - разуме разлику између клијентских и серверских скрипт језика - разуме појам и начин функционисања .NET Framework библиотеке класа - познаје основе ASP.NET скрипт језика - познаје основе Java Script језика - познаје Internet Information Services (IIS) и како функционише веб сервер - разуме објектни модел XML документа - разуме инфраструктуру XML веб сервиса - разуме појам и начин функционисања MVC Framework архитектуре 	<ul style="list-style-type: none"> - коришћењем HTML језика форматирају текст, подешава позадину, убацује слике и хиперлинкове и у целости планира и подешава изглед веб странице - прави обрасце на веб страници - користи CSS кодове за прилагођење изгледа елемената на веб страници - поставља презентацију на сервер - креира веб форме, поставља и користи серверске и HTML веб контроле - користи контроле за рад са подацима - креира функције и користи догађаје прозора, миша, обрасца, тастера. - креира и конфигурише виртуелни директоријум - инсталира и конфигурише Apache сервер - користи класе и функције за рад са XML подацима веб - поставља и конфигурише XML веб сервисе на серверу - креира MVC моделе, контролере, погледе 	
Израда и управљање базом података	<ul style="list-style-type: none"> - објашњава области примене информационих система - објашњава процес моделовања система - дефинише појам базе података - разуме основне концепте и структуру релационих база података - познаје системе за управљање базама 	<ul style="list-style-type: none"> - анализира захтеве корисника и израђује (црта) једноставне ЕР дијаграме - преводи објекте из ЕР дијаграма у релациони модел - користи алате за моделирање - предлаже измене релационог модела како би се достигла виша нормална форма 	

Стручне компетенције	Знања	Вештине	Способности и ставови
	<ul style="list-style-type: none"> - података - разуме последице лошег дизајна базе података 	<ul style="list-style-type: none"> - израђује структуру једноставне базе података - врши измену структуре базе података - користи упитни језик за приступ подацима базе података - осмисли решење постављеног захтева коришћењем функција, процедура и курсора 	
Израда техничке документације	<ul style="list-style-type: none"> - наведе основну намену и сврху техничке документације - наведе основне елементе техничке документације - објасни животни циклус и фазе пројекта - наведе основну намену и сврху идејног решења - наведе шта треба да садржи квалитетно техничко решење - наведе начине за контролу извршених послова 	<ul style="list-style-type: none"> - познаје рад са програмима за пројектовање техничке документације - планира софтверске компоненте у циљу компромиса цене и квалитета софтвера - дефинише предмет и предрачун у оквиру техничке документације - врши контролу извршених послова - саставља записник примедби 	

АНЕКС 2. ЗБИРКА ТЕОРИЈСКИХ ЗАДАТАКА

РАДНА ВЕРЗИЈА

Драги ученици,

Пред вама је збирка задатака за завршно тестирање у оквиру матурског испита за образовни профил Електротехничар информационих технологија. Збирка је намењена вежбању и припремању за полагање испита за проверу стручно теоријских знања, и то из стручних предмета: **Програмирање, Веб дизајн, Веб програмирање, Базе података**. У збирци се налазе задаци који ће бити на тесту у потпуно истој или делимично измененој форми.

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 максималан број бодова који доноси.

Тест који ћете решавати на матурском испиту садржи задатке свих нивоа сложености којима се испитује оствареност исхода образовања за образовни профил Електротехничар информационих технологија. 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. Код рачунских задатака, ако их има, тачан одговор се признаје само уз приказан поступак решавања. Збирка задатака не садржи решења.

Збирку задатака су израдили тимови наставника из школа у Републици Србији у којима се реализује матурски испит школске 2021/2022. године за образовни профил Електротехничар информационих технологија у сарадњи са стручњацима Завода за унапређивање образовања и васпитања.

Желимо вам срећан и успешан рад!

Аутори

РАДНА ВЕРСИЈА

ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК С

У следећим задацима заокружите број испред траженог одговора

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

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

1

2. 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

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

3. 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

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

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

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

typedef struct
{
    Tacka3D centar;
    float R;
} Lopta;
```

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

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};`

5. 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

6. 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

7. Дата је наредба декларације, а затим и наредба форматированог 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 x= 0
2. x = 5.560000 x = 5.560000e+000
3. x = 5.560000 x = 5.56
4. x = 5.56 x = 5.560000e+000

8. 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

9. 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**

10. Наредбом у Ц језику треба проверити да ли је број паран или непаран. Проценити која од датих наредби врши ову проверу и заокружити број испред тачно написане наредбе.

2

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!!");

11. Код дат у тексту задатка треба реализовати помоћу једне if наредбе. Заокружити број испред понуђеног тачног одговора:

2

```
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;
```

12. Дат је део кода на програмском језику Ц:

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

2

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++];

13. 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

14. Дата је декларација променљивих `pod, br` и део кода у програмском језику Ц:

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

2

Закључити шта представља вредност коју променљива `br` добије извршењем кода. 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`

15. Дат је део кода на програмском језику Ц, који контролише унос целобројне променљиве `n`. Одредити вредности које променљива `n` може добити. Заокружити број испред траженог одговора:

```
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

-
16. 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;
```

2

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 }

-
17. 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) { ... }
```

2

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

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);

-
18. 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);
```

2

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

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);

19. 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);  
}
```

2

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

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

20. 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);

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

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

У следећим задацима заокружите бројеве испред тражених одговора

22. Заокружити бројеве испред **ТАЧНИХ** исказа који се односе на дефиницију while циклуса:
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).
23. 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.
24. 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
25. Дате су наредбе декларације променљивих (са и без иницијализације вредности) написане на програмском језику Ц. Заокружити бројеве испред исправно написаних наредби декларације променљивих:
1. `int a=b=c=5;`
 2. `int a=5, b=5, c=5;`
 3. `char zn="a";`
 4. `long a; b=5; c;`
 5. `int a=0xf2;`
 6. `char zn='\\b';`
26. Декларисане су следеће променљиве:
- ```
float x, z;
const float y;
```
- Заокружити бројеве испред **НЕИСПРАВНО** написаних наредби доделе вредности променљивама:
1. `x %= y;`
  2. `x =+ 5;`
  3. `x += y + 5;`
  4. `x =/ y + 5;`
  5. `y = x + z;`
  6. `x = z = y + 5;`

27. Дата је наредба декларације **int a, b;**  
Имајући у виду дату декларацију, заокружити бројеве испред НЕИСПРАВНО написаних наредби форматираног уноса података:

1. `scanf ("%d%f", &a, &b);`
2. `scanf ("%d%d", &a, &b);`
3. `scanf ("%d%d", &a, &b);`
4. `scanf ("%d%d", a, b);`
5. `scanf ("%d%d", &a);`
6. `scanf ("%5d%5d", &a, &b);`

1,5

28. 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:

1. `int a[10]={1,2,3};`
2. `int a[5]={-3, -2, -1, 0, 1, 2, 3};`
3. `int a[]={10,20,30,40,50};`
4. `int[5] a={1, 2, 3, 4, 5};`
5. `int a={10,20,30,40,50};`
6. `int a[5]={'1', '2', '3', '4', '5'};`

1,5

29. 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:

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

30. 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

31. У следећем задатку заокружити бројеве испред тражених одговора.  
Дата је наредба у Ц језику, која температуру у Целзијусима **tempc** претвара у температуру у Фаренхајтима **tempf**. Подаци tempc и tempf су реални бројеви обичне тачности. Проценити који изрази дају тачно решење.

1. **temf = (9 / 5) \* temc + 32;**
2. **temf = 9 / 5 \* temc + 32;**
3. **temf = 9 \* temc / 5 + 32;**
4. **temf = 32 + 9 \* temc / 5;**

2

32. Дата је if-else наредба:

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

Заокружити бројеве испред понуђених switch наредби које су еквивалентне датој if-else наредби:

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

33. У програму на програмском језику Ц извршена је следећа декларација, а касније и резервација меморијског простора за низ реалних бројева обичне тачности дужине n:

```
float *B;
int n;
B=(float*)calloc(n,sizeof(float));
```

Заокружити бројеве испред исправно написаних наредби за **ПРИКАЗ** i-тог елемента низа B:

1. **printf("%f", B[i]);**
2. **printf("%f", &B[i]);**
3. **printf("%f", B+i);**
4. **printf("%p", \*(B+i));**
5. **printf("%f", \*(B+i));**

2

- 
34. У програму на програмском језику Ц извршена је следећа декларација, а касније и резервација меморијског простора за низ реалних бројева обичне талности дужине n:

```
float *B;
int n;
B=(float*)calloc(n,sizeof(float));
```

Заокружити бројеве испред исправно написаних наредби за УНОС i-тог елемента низа B:

2

1. scanf("%f", B[i]);
2. scanf("%f", B+i);
3. scanf("%p", B+i);
4. scanf("%f", &B[i]);
5. scanf("%f", \*(B+i));

- 
35. 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

- 
36. 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...**

2

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

- 
37. 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:

2

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);

- 
38. У програмском језику С 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

- 
39. 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

- 
40. 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:

3

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]);

41. 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:

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);

3

### Допуните следеће реченице и табеле

42. 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

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 = \_\_\_\_\_

2

43. Дати је декларација променљивих **int a=3, b=15;**

Израчунати вредност коју ће променљиве имати по извршењу следеће наредбе:

```
b %= ++ a;
```

2

a = \_\_\_\_\_

b = \_\_\_\_\_

44. 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 = \_\_\_\_\_

- 
45. У програмском језику Ц декларисане су две целобројне променљиве:

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

За дате вредности променљиве **izbor**, одреди вредност променљиве **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;
}
```

1. за izbor=3, x=\_\_\_\_\_

3. за izbor=4, x=\_\_\_\_\_

2. за izbor=10, x=\_\_\_\_\_

4. за izbor=2, x=\_\_\_\_\_

2

- 
46. 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;
```

- 
47. Дата 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

48. 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:

```
for (k=0; k< _____; k++)
```

```
mat [_____] [_____] *=2;
```

2

49. 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

3.  $-10 \% 3 =$  \_\_\_\_\_

4.  $10. \% 5 =$  \_\_\_\_\_

5.  $10 \% (-3) =$  \_\_\_\_\_

6.  $(100/3) \% 6 =$  \_\_\_\_\_

50. 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 = \_\_\_\_\_;**

1.  $p[0]$

**for (i=n-2; i \_\_\_\_\_; \_\_\_\_\_)**

2.  $p[n-1]$

**\_\_\_\_\_ = \_\_\_\_\_;**

3.  $p[n]$

**\_\_\_\_\_ = pom;**

4.  $p[i+1]$

**\_\_\_\_\_ = \_\_\_\_\_;**

5.  $p[i-1]$

**\_\_\_\_\_ = pom;**

6.  $p[i]$

3

7.  $i++$

8.  $i--$

9.  $>=0$

10.  $>0$

51. Дата 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

52. У програмском језику Ц, дате су следеће декларације: `int A[50], i, n;`

Потребно је формирти вектор са следећим вредностима:

| <code>i=0</code> | <code>i=1</code> | <code>i=2</code> | <code>i=3</code> | <code>i=4</code> | <code>i=5</code> | <code>...</code> | <code>i=n-1</code> |
|------------------|------------------|------------------|------------------|------------------|------------------|------------------|--------------------|
| 1                | 2                | 4                | 7                | 11               | 16               | ...              | ???                |

Допунити програмски код којим се формира овај вектор:

`A[0]=1;`

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

3

53. 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

54. 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

55. 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

56. 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);
```

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

**a** = \_\_\_\_\_

**b** = \_\_\_\_\_

**c** = \_\_\_\_\_

3

57. 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 = \_\_\_\_\_

### У следећим задацима уредите и повежите појмове према захтеву

58. 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 погља 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

59. Декларисана је реална променљива `float w=123.456;:`

Са леве стране су дати различити прикази вредности пременљиве добијени коришћењем наредби форматираних излаза које су приказане са десне стране. Поред сваке наредбе, на предвиђену линију уписати рени број приказа добијеног изршавањем те наредбе:

1. 123.456000 \_\_\_\_\_ `printf("%g", w);`  
2. 1.234560e+002 \_\_\_\_\_ `printf("%f", w);`  
3. 123.456 \_\_\_\_\_ `printf("%.2f", w);`  
4. 123.46 \_\_\_\_\_ `printf("%e", w);`

2

60. Са леве стране дати су математички изрази, а са десне запис израза на програмском језику Ц. На линију испред записа израза, унети 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

61. 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 с (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 \_\_\_\_\_ 'a'
6. Duz \_\_\_\_\_ 'b'

62. 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

63. 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:

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

2

64. 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:

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

2

65. 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:

- |                         |       |                                                                          |
|-------------------------|-------|--------------------------------------------------------------------------|
| ftell(dat)              | _____ | 1. позиционирање на почетак датотеке                                     |
| fseek(dat, 0, SEEK_END) | _____ | 2. позиционирање на крај датотеке                                        |
| fseek(dat, 0, SEEK_SET) | _____ | 3. одређује позицију у датотеци у виду броја байтова од почетка датотеке |
| rewind(dat)             | _____ | 4. ништа од понуђеног                                                    |

2

66. Са леве стране 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 (u dekadnom obliku)                       |
| 3. %s  | _____ | long                                                 |
| 4. %ld | _____ | unsigned                                             |
| 5. %f  | _____ | signed int (dekadni, heksadekadni ili oktalni oblik) |
| 6. %e  | _____ |                                                      |
| 7. %hd | _____ |                                                      |
| 8. %u  | _____ |                                                      |
- 2,5
67. 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 |
- 2,5
68. 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)          |
| 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) |
- 3
69. 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?                 | _____ |            |
- 3

70. 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"<br>5. "sagain"<br>6. "myM8sagain" |

4

РАДНА ВЕРЗИЈА

## ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК С#

### У следећим задацима заокружите број испред траженог одговора

71. Дати су типови променљивих у програмском језику C#. 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

72. Одредити какви могу бити чланови класе (поља и методе) у програмском језику C#.

Заокружити број испред очекиваног одговора:

1. Локални и глобални
2. Процедурални и непроцедурални
3. Статички (класни) и нестатички (објектни)
4. Спољашњи и унутрашњи

1

73. 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

74. 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

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

75. Заокружити број испред исказа који представља исправан наставак дате тврдње:

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

76. 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

77. 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 никде не позива.
3. Програм се нормално извршава и приказује 17 једанпут.
4. Програм се нормално извршава и приказује 17 двапут.

78. Дат је код програма у програмском језику C# који формира и штампа елементе низа **a**.

Анализирати дати код и проценити шта ће се дододити 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.Write(a[i] + " ");
 }
 }
}
```

Заокружити број испред очекиваног одговора:

2

1. Програм приказује бројеве 0 1 2 3 4 на екрану.
2. Програм има грешку, јер ће у последњој наредби **Console.Write** 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.Write** у методу **Main** неће имати дефинисану вредност.

- 
79. У програмском језику C# 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:

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

- 
80. У програмском језику C# дата је декларација једне стринг и једне целобројне променљиве, као и део кода:

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

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

1. Primer6665  
131Primer
2. Primer6665  
6665Primer
3. Primer131  
131Primer
4. PrimerBA  
BAPrimer

2

- 
81. 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:

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(...).

2

82. 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);
 }
 }
 }
}
```

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 ....

83. 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]

84. У програмском језику C# дат је рекурзивни метод који проверава да ли је неки string 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 slučaj
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)

85. У програмском језику C# 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. }
```

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)

86. Дат је код програма у програмском језику C#. 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);
 }
 }
}
```

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.

87. 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. }
```

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.

2

2

2

88. 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. }
```

Заокружити број испред очекиваног одговора:

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

2

89. 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){
 KlasaA a1 = new KlasaA ();
 KlasaA a2 = new KlasaA ();
 Console.WriteLine(a1.Equals(a2));
 }
}
class KlasaA {
 int x;
 public bool Equals(KlasaA a) {
 return this.x == a.x;
 }
}
```

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.

2

90. 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 KlasaA();
 Object a2 = new KlasaA();
 Console.WriteLine(a1.Equals(a2));
 }
}
class KlasaA {
 int x;
 public bool Equals(KlasaA 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 А 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.

91. 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.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)`.

92. 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 (ArithmetricException ae)
{
 Console.WriteLine(" Arithmetric Exception");
}
Console.WriteLine("finished");
```

Заокружи број испред тачног одговора:

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

2

93. 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 изведених 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 основна класа „маскира“

2

#### У следећим задацима заокружите бројеве испред тражених одговора

94. Дати су искази 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

- 
95. Да би наслеђени метод могао да се **редефинише** и тиме измени 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

- 
96. 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

- 
97. 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 | 1. penzija |
| 2. adresa  | 2. adresa  |
| 3. ime     | 3. ime     |
| 4. plata   | 4. plata   |
| 5. struka  | 5. struka  |
| 6. razred  | 6. razred  |

1,5

98. 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** креиран је објекат **s** класе **Sin** (`Sin s = new Sin();`) Заокружити бројеве испред поља која ће бити видљива у креираним 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
2. adresa
3. ime
4. plata
5. struka
6. firma
7. prosek
8. razred
9. skola

1,5

99. 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;

- 100 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

- 101 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

- 102 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)`.

- 103 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**.

2

- 104 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 класе
3. Својства (**property**)
4. Приватни чланови класе
5. Конструктор класе

2

- 105 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 A је " + x; }
 }
}
```

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***.

- 106 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{ }
}
```

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***

2

2

- 
- 107 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

### Допуните следеће реченице и табеле

- 
108. 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 \_\_\_\_\_

Наредбе које се извршавају и ако се деси и ако се не деси  
грешка стављају се унутар блока \_\_\_\_\_

1,5

Наредбе које могу узазвати грешку стављају се унутар блока \_\_\_\_\_

- 
109. 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, креирани су три објекта ових класа на следећи начин:

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

3

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

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

110. 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:

```
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(); \_\_\_\_\_

111. 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:

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 \_\_\_\_\_

112. 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:

```
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(); \_\_\_\_\_

3

3

4

## У следећим задацима уредите и повежите појмове према захтеву

113. 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 |
| 3. partial   | _____ | Класа која се не може инстанцирати                                      |
| 4. interface | _____ | Класа из које се не може наслеђивати                                    |

2

114. 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 изведенних класа |
| 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                           |

2

115. 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 елемент класе:

- |                        |       |                                     |
|------------------------|-------|-------------------------------------|
| 5. поље (атрибут)      | _____ | Опис функционалности објеката класе |
| 6. деструктор          | _____ | Контрола приступа пољима класе      |
| 7. конструктор         | _____ | Опис особина објеката класе         |
| 8. метод               | _____ | Креирање објеката класе             |
| 9. својство / property | _____ | Уништавање објеката класе           |

2,5

## ПРОГРАМИРАЊЕ – ПРОГРАМСКИ ЈЕЗИК ЈАВА

### У следећим задацима заокружите број испред траженог одговора

116. 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

1

117. 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

1

118. 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

119. 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

120. Заокружити број испред исказа који представља исправан наставак дате тврдње:

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

121. 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

122. 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); }
}
```

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 никде не позива.
3. Програм се нормално извршава и приказује 17 једанпут.
4. Програм се нормално извршава и приказује 17 двапут.

123. Дат је код програма у програмском језику Јава који формира и штампа елементе низа ***a***. Анализирати дати код и проценити шта ће се дододити 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.

- 
124. 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:

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

- 
125. 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:

2

1. Primer6665  
131Primer
2. Primer6665  
6665Primer
3. Primer131  
131Primer
4. PrimerBA  
BAPrimer

- 
126. 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:

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(...).

2

127. 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);
 }
 }
}
```

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 ....

2

128. 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 наредбу.

```
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())

129. 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 slučaj
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)

130. 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. }
```

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)*

2

131. 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);
 }
}
```

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.

132. 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.

133. 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. }
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);

134. 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 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.

135. 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;
 }
}
```

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.

2

136. 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)`.

- 
137. 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 (ArithmaticException ae)
{
 System.out.println(" Arithmatic Exception");
}
System.out.println("finished");
```

Заокружки 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: Arithmatic Exception

2

- 
138. 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 осавна класа „маскира“

2

### У следећим задацима заокружите бројеве испред тражених одговора

- 
139. 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

- 
- 140 Да би наслеђени метод могао да се **редефинише** и тиме измени његова функционалност у класама наследницама, 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

- 
- 141 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

- 
- 142 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 креираним објектима 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

- 143 Дати су изкази који се односе на правила писања try-catch-finally блокова за руковање изузетцима. 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

- 144 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) { _____; }
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;

- 145 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

- 146 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

- 147 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()");
 }
}
```

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. **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)**.

- 148 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**.

2

- 149 Заокружити бројеве испред наведених чланова класе који који се ни под којим условима **НЕ** наслеђују из родитељске класе у изведену класу:

1. Final поља
2. Заштићени чланови класа
3. Сетери и гетери
4. Приватни чланови класа
5. Конструктор класе

2

- 150 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 u A je " + 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 `A`.
4. Приликом извршавања наредбе `System.out.println(obj)`, програм позива метод `toString()` из класе `Object`.

- 151 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

- 
- 152 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

### Допуните следеће реченице и табеле

- 
153. 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 \_\_\_\_\_

Наредбе које се извршавају и ако се деси и ако се не деси грешка стављају се унутар блока \_\_\_\_\_

1,5

Наредбе које могу изазвати грешку стављају се унутар блока \_\_\_\_\_

- 
154. 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, креирани 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 објектата:

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

- 
155. 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:

```
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(); \_\_\_\_\_

3

- 
156. 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

157. 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, креирани су два објекта ових класа на следећи начин:

```
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(); \_\_\_\_\_

4

#### У следећим задацима уредите и повежите појмове према захтеву

158. 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  
Класа садржи само декларације метода, али не и дефиницију (тело) методе
2. abstract \_\_\_\_\_
3. final \_\_\_\_\_ Мора бити креирана као унутрашња (inner) класа
4. interface \_\_\_\_\_ Класа која се не може инстанцирати

2

159. 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. 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

2

160. Са леве стране су наведени делови/елементи класе, а са десне стране улоге појединих класних елемената. На линију испред описа улоге унети редни број под којим је наведен одговарајући елеменат класе:

- |                           |       |                                     |
|---------------------------|-------|-------------------------------------|
| 1. атрибут                | _____ | Опис функционалности објеката класе |
| 2. garbage collector      | _____ | Контрола приступа пољима класе      |
| 3. конструктор            | _____ | Опис особина објеката класе         |
| 4. метод                  | _____ | Креирање објеката класе             |
| 5. методе сетери и гетери | _____ | Уништавање објеката класе           |

2,5

РАДНА ВЕРЗИЈА

## ВЕБ ДИЗАЈН

### У следећим задацима заокружите број испред траженог одговора

161. Заокружити број испред исказа који тачно дефинише 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 мултимедијални садржај састављен у једну целину
3. веб сајт чини интернет прегледач - програм који се користи за читање хипертекст докумената
4. интернет сервис (www) који омогућава корисницима размену докумената која се често срећу од текста, слика и мултимедијалних компоненти

1

162. Заокружити број испред тачног одговора.

Основни протокол који користи веб сервер је:

1. SMTP
2. HTTP
3. HTML
4. WWW

1

163. Заокружити број испред тачног одговора. 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

1

164. Заокружити број испред тачног одговора.

Унутар HTML странице, CSS правила се креирају навођењем селектора елемента, својства и вредности. Како се дефинишу селектори HTML елемента:

1. Селектори се дефинишу искључиво на основу имена HTML елемента
2. Селектори се дефинишу на основу имена елемента, назива класе или идентификатора елемента
3. Селектори се дефинишу на основу атрибута сваког HTML елемента
4. Селектори се дефинишу коришћењем кључне речи selector и атрибута

1

165. Заокружити број испред тачног одговора.

Дефинисан је стил елемента у оквиру кога се налази позадинска слика. Коју вредност својство `background-repeat` треба да има уколико слика не треба да се понавља у оквиру елемената:

1. `background-repeat: fixed;`
2. `background-repeat: null;`
- 3. `background-repeat: no-repeat;`**
4. `background-repeat: repeat-x;`

1

166. Заокружити број испред тачног одговора.

Селектором `a:visited` омогућава се стилизовање:

1. активног линка
- 2. посјећеног линка**
3. линка када се преко њега пређе мишем
4. елемента са називом класе: a

1

167. Заокружити број испред тачног одговора.

Које од понуђених својстава 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

168. Заокружити број испред тачног одговора.

Аналазирати следећи HTML код и одабрати која 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 искошено**
4. Неће се применити никакав ефекат на текст

1

169. Заокружити број испред линије кода којом се у HTML страницу укључује екстерна CSS датотека `style.css` 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>`

2

170. Заокружити број испред тачног одговора. Који је од наведених начина за укључивање CSS стила најпогоднији за стилизовање, а посебно за касније одржавање и ажурирање HTML странице:

1. Увежена екстерна CSS датотека преко хипервезе у заглављу HTML веб странице - External Style Sheet
2. Дефинисан интерни CSS стил у оквиру head секције `<style type="text/css">...</style>` - Internal Style Sheet
3. Дефинисан стил у оквиру елемента HTML стране - Inline style
4. Не постоји решење за укључивање CSS стила у стилизовање HTML странице

2

171. Заокружити број испред тачног одговора.

Дефинисан је HTML елемент:

```
<div id="container"> Maturski ispit - EIT</div>
```

Које CSS правило дефинише стил за дати елемент:

1. `container { font-size: 1.5em; }`
2. `#container { font-size: 1.5em; }`
3. `.container { font-size: 1.5em; }`
4. `selector container { font-size: 1.5em; }`

2

172. Заокружити број испред тачног одговора.

Уколико је стил једног DIV елемента дефинисан са три класе blue, black и top, заокружити линију кода којом је елемент исправно дефинисан у оквиру HTML стране:

1. `<div id="blue black top">Maturski ispit -EIT</div>`
2. `<div id="blue" class="black" class="top">Maturski ispit -EIT</div>`
3. `<div class="blue" class="black" class="top">Maturski ispit - EIT</div>`
4. `<div class="blue black top">Maturski ispit -EIT</div>`

2

- 
173. Заокружити број испред тачног одговора.

Дато је својство и додељена му је вредност:

***background-attachment: fixed;***

Закључити који од понуђених исказа описује дефинисани стил елемента са позадинском сликом:

2

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

- 
174. Заокружити број испред траженог одговора.

Дат је следећи код који стилизује три елемента странице 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

Проценити шта од понуђеног описује ефекат извршења горњег кода:

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**

## У следећим задацима заокружите бројеве испред тражених одговора

- 175 Заокружите тачне исказе. 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
- 176 Заокружити бројеве испред тражених одговора.
- Шта од наведеног представља скуп специјализованих програма са функцијом веб сервера:
1. Microsoft Internet Information Services
  2. Microsoft SQL Server
  3. Apache Web Server
  4. Microsoft NT Server
- 177 Међу понуђеним исказима заокружити оне које се односе на динамичке веб странице:
1. Могу приступити базама података
  2. Странице се сastoјe искључиво од 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. Ажурирање података на сајту је компликованије и спорије него код статичког сајта
- 178 Заокружити могуће начине укључивања 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>`

## Допуните следеће реченице и табеле

179. 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

<link rel="stylesheet" type="text/css" href="style.css">

## У следећим задацима уредите и повежите појмове према захтеву

180. Повезати појмове према захтеву.

Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML таг повезује са одговарајућим индикатором који ће се видети на HTML страницама:

- |                           |   |                             |
|---------------------------|---|-----------------------------|
| 1. <font color="#ff0003"> | 3 | приказ назива трећег нивоа  |
| 2. <b>                    | 4 | приказ текста дате величине |
| 3. <h3>                   | 1 | приказ текста у боји        |
| 4. <font size="3">        | 2 | приказ подебљаног текста    |

1,5

181. Повезати појмове према захтеву.

Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML таг повезује са одговарајућим индикатором који ће се видети на HTML страницама:

- |                                 |   |                             |
|---------------------------------|---|-----------------------------|
| 1. <ol>                         | 2 | дефинисање елемента листе   |
| 2. <li>                         | 3 | постављање позадинске боје  |
| 3. <body bgcolor="#ffff00">     | 4 | постављање позадинске слике |
| 4. <body background="0001.jpg"> | 1 | дефинисање нумерисане листе |

1,5

182. Повезати појмове према захтеву.

Са леве стране су дати нумерисани HTML тагови, а са десне су наведени индикатори. На линију уписати број којим се HTML таг повезује са одговарајућим индикатором који ће се видети на HTML страницама:

- |                           |   |                       |
|---------------------------|---|-----------------------|
| 1. <tr>                   | 4 | Дефинисање хиперлинка |
| 2. <td>                   | 1 | Нови ред у ћелији     |
| 3.  | 2 | Нова ћелија у табели  |
| 4. <a href="index.html" > | 3 | Уметање слике         |

183. Са леве стране су дати интернет сервиси, а са десне стране су дефинисане операције које се извршавају помоћу тих сервиса.

На линију испред дефинисане операције написати број њему одговарајућег сервиса.

1. E-mail	4	успостављање везе са удаљеним рачунаром и рад на њему	2
2. FTP	3	приказ HTML веб страница	
3. WWW	1	слање електронске поште	
4. Telnet	2	пренос датотека са удаљеног сервера	

184. Повезати појмове према захтеву.

HTML документ може да прими податке од корисника помоћу форми (формулара). Повежите дате тагове и атрибуте са њиховим дефиницијама:

1. FORM	3	Дефинише где проследити податке са forme после предаје (submit) forme	2
2. INPUT	4	Одређује начин на који се подаци са forme шаљу на дефинисано одредиште (може бити „post“ или „get“)	
3. ACTION	1	Основни таг формулара са којим се креира формулар за унос података од стране корисника	
4. METHOD	2	Дефинише поље за унос податка унутар HTML forme.	

185. Направите редослед потребних корака за приказ статичке веб странице уносом редних бројева на линије испред описа корака, почев од броја 1 до броја 5.

4	Веб сервер шаље пронађену страну клијенту - веб претраживачу.
1	Корисник захтева да види веб страну (кликом на линк, укуцавањем адресе у адресну линију веб претраживача и слично)
3	Аутор је креирао страну која се састоји од HTML кода и ставио је на веб сервер.
5	Веб претраживач обрађује добијени HTML код и приказује кориснику уредно форматирану страницу са свим елементима (сликама, линковима, табелама, ...)
2	Веб сервер проналази HTML страницу коју је корисник захтевао

186. Одредите редослед потребних корака за приказ динамичке веб странице уносом редних бројева (почев од 1) на линије испред описа корака.

- |   |                                                                                                                                                                  |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6 | Веб претраживач обрађује добијени HTML код и приказује кориснику страницу са свим елементима.                                                                    |
| 4 | Веб сервер извршава програмски код који је саставни део стране и креира HTML код.                                                                                |
| 1 | Корисник креира захтев за преглед динамичке веб странице. Захтев се прослеђује од клијента (веб претраживача) до веб сервера на коме се налази захтевана страна. |
| 3 | Аутор је креирао страну која се састоји од серверских контрола и инструкција у неком програмском језику и ставио је на веб сервер.                               |
| 2 | Веб сервер обрађује захтев корисника и проналази динамичку страну коју је корисник захтевао                                                                      |
| 5 | Веб сервер шаље преко Интернета генерисани HTML код веб претраживачу.                                                                                            |

187. Уредите појмове према захтеву.

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:

- |   |         |
|---|---------|
| 5 | </body> |
| 2 | <head>  |
| 6 | </html> |
| 1 | <html>  |
| 3 | </head> |
| 4 | <body>  |

## ВЕБ ПРОГРАМИРАЊЕ

### У следећим задацима заокружите број испред траженог одговора

188. Заокружити исказ који описује улогу Domain Name Server-a:

1. превођење имена домена у IP адресу
2. хостовање веб сајта
3. вршење функције главног чвора у локалној рачунарској мрежи
4. приказ динамичких веб страница

1

189. Једну од платформи за развој веб апликација развио је и Microsoft. Заокружити назив Microsoft-ове платформе за развој веб апликација:

1. HTML
2. JSP
3. PHP
4. ASP.NET

1

190. 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

191. Заокружити од понуђених опција који сумбол се користи за коментаре у ASP.NET MVC Razor синтакси:

1. //
2. /\* ... \*/
3. <!--...-->
4. @\* ... \*@

1

192. SOAP протокол (Simple Object Access protocol) који се користи за размену података између рачунара коришћењем веб сервиса, у основи користи један скрипт језик. Заокруживањем редног броја, обележити о ком језику се ради.

1. HTML
2. CSS
3. JavaScript
4. XML

1

193. Заокружити исказ који дефинише улогу Proxy сервера:
1. Омогућава пренос датотека са удаљеног рачунара и ка 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
194. Заокружити исказ који допуњује опис понашања променљивих током извршења програма написаног у језику JavaScript. *Током извршавања апликације написане у JavaScript-у:*
1. Није могуће мењати типове променљивих у току извршавања апликације
  2. **Типови променљивих се могу мењати током извршавања програма**
  3. Типови променљивих се обавезно мењају током извршавања апликације у одговарајући веб тип променљиве
  4. JavaScript не подржава типове променљивих
195. 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");`
196. Заокружити један од понуђених типова сервиса који може бити хостован у конзолној или десктоп апликацији:
1. ASMX
  2. RESTful
  3. **WCF**
  4. XML

#### У следећим задацима заокружите бројеве испред тражених одговора

- 197 Заокружите број испред назива серверског скрипт језика (серверске веб технологије) за израду динамичких веб страница:
1. **PHP**
  2. jQuery
  3. **ASP.NET**
  4. **JSP**
  5. VBScript
  6. CSS
  7. HTML

198	У клијентском скрипту језику Java Script постоји неколико могућности конверзије стринга у број. Заокружити функције које омогућавају те конверзије:	<ol style="list-style-type: none"> <li>1. Није могуће мењати типове променљивих у току извршавања апликације</li> <li><b>2. Функција EVAL - процењује стринг и ако је могуће претвара га у број</b></li> <li><b>3. Функција parseInt – конвертује стринг у целобројни број, ако је могуће</b></li> <li><b>4. Функција parseFloat – конвертује стринг у реалан број, ако је могуће</b></li> <li>5. Функција parseDouble – конвертује стринг у реалан број, ако је могуће</li> <li>6. Функција tryParseInt – конвертује стринг у целобројни број, ако је могуће</li> </ol>	1,5
199	Обележити шта од наведеног је дефинисано WSDL језиком (Web Services Description Language) (заокружити више понуђених одговора):	<ol style="list-style-type: none"> <li><b>1. Комуникациони интерфејс за веб сервис</b></li> <li>2. Начин имплементације метода веб сервиса</li> <li><b>3. Списак метода веб сервиса</b></li> <li>4. Комуникациони протокол за веб сервис</li> </ol>	2
200	Microsoft .NET Framework садржи базне класе које пружају широк спектар могућности. Заокружити елементе који су укључени у .NET Framework:	<ol style="list-style-type: none"> <li>1. класе корисничког интерфејса</li> <li><b>2. класе за приступ подацима и базама</b></li> <li>3. класе корисника</li> <li>4. веб сервер и примере базе података</li> <li><b>5. класе за манипулацију XML докумената</b></li> <li>6. скрипт језик који се извршава на клијент страни</li> </ol>	3
201	Повратне вредности акције MVC контролера (controller action method) могу бити различитих типова. Заокружити бројеве испред могућих повратних вредности.	<ol style="list-style-type: none"> <li><b>1. ViewResult</b></li> <li>2. MVCResult</li> <li>3. ModelResult</li> <li><b>4. JsonResult</b></li> <li><b>5. RedirectResult</b></li> <li>6. ActionResult</li> </ol>	3
202	XML Schema (XSD) опсује структуру XML документа Обележити шта од наведеног дефинише XML Schema (заокружити више понуђених одговора):	<ol style="list-style-type: none"> <li><b>1. Типове података XML елемената и атрибута</b></li> <li>2. Вредности XML елемената и атрибута</li> <li><b>3. XML елементе који представљају child (деца) елементе</b></li> <li><b>4. Уређење child елемената</b></li> <li>5. Начин сортирања атрибута у оквиру елемената</li> <li>6. Међусобни однос корених (root) елемената документа</li> </ol>	3

## Допуните следеће реченице и табеле

203. Дата је MVC стандардна ruta (default route)

`http://localhost/Customer/Details/5`

која има 3 сегмента. Препознати на основу дате руте вредности ових сегмената и допунити реченицу:

1,5

Име контролера (Controller Name) је: Customer, назив методе (Action Method Name) је:  
Name: Details а ID параметра методе је дат са: 5.

## У следећим задацима уредите и повежите појмове према захтеву

204. У JavaScript језику свака веб страница има објекте наведене у левој колони. На десној страни дати су описи наведених објеката. Повезати објекат са одговарајућим описом уносом редног броја којим је објекат нумерисан на означену линију:

1. window	3	садржи URL адресе које су претходно биле посећене, као и URL адресе које су посећене након посете тренутној страници
2. location	4	представља тренутно учитани документ и служи као приступна тачка садржају странице
3. history	2	садржи информације о URL адреси документа који је тренутно приказан у посматраном прозору
4. document	1	објекат највишег нивоа који садржи све друге објекте и представља прозор претраживача

205. У JavaScript језику многи објекти имају уграђене функције (методе) које симулирају догађаје настале услед акција корисника. Уносом редног броја методе (лева колона) на предвиђену линију, повежите метод и одговарајућу акцију корисника.

1. focus	4	Изађе из фокуса елемента форме
2. submit	3	Учита страницу у прегледач
3. load	1	Уђе у фокус неког елемента форме
4. blur	2	Изврши слање форме

206. Веб обрасци могу да садрже различите типове компоненти. Редни број под којим је наведена категорија компоненти уписати на линију испред одговарајуће групе компоненти веб образца:

1. HTML контроле	3	TextBox, Label, Button, ListBox, DropDownList, DataGridView
2. Контроле за податке	4	FileSystemWatcher, EventLog, MessageQueue
3. Серверске контроле	2	SqlConnection, SqlCommand, OleDbConnection
4. Системске компоненте	1	Text Area, Table, Image, Submit Button, Reset Button

207. Садашње верзије IIS сервера изграђене су на модуларној архитектури. Повежите називе модула са функцијама које обављају:

1. HTTP модули	3	Модули за праћење и извештавање о догађајима насталим током процесирања захтева...
2. Безбедностни модули	4	Модули за обраду захтева за статичке фајлове, враћање подразумевне странице када клијент не наведе ресурс у захтеву, излиставање садржаја директоријума...
3. Модули за евиденцију и дијагностику	1	Модули за одговорање на информације, враћање HTTP грешака, преусмеравање захтева...
4. Модули садржаја	2	Модули за ауторизацију УРЛ адреса, филтрирање захтева...

## БАЗЕ ПОДАТАКА

У следећим задацима заокружите број испред траженог одговора

208. Међу понуђеним ентитетима, одредити ентитет са атрибутима који **НИСУ** одговарајући.  
Заокружити број испред траженог одговора:

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

1

209. 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

210. Заокружити број испред траженог одговора.

Одредити оператор који би требало користити у WHERE клаузули SELECT наредбе да би били приказани само они ученици чије презиме почиње словом A:

1. IN
2. LIKE
3. BETWEEN
4. IS LIKE
5. BEGINS WITH

1

211. Заокружити број испред траженог одговора.

Табела UCENICI поред осталих података, садржи и вредност стипендије. Одредити оператор који треба употребити у WHERE клаузули SELECT наредбе да би били приказани сви ученици код којих није позната и није унета вредност у колону **stipendija**:

1. =NULL
2. ISNULL
3. ==NULL
4. IS NULL
5. LIKE NULL

1

212. Дат је упит:

```
SELECT *
FROM ucenici
WHERE odeljenje=4 OR odeljenje=7 OR odeljenje=10
```

Заокружити оператор који треба користити у датом упиту да би се избегло вишеструко коришћење оператора OR:

1. LIKE
2. BETWEEN
3. AND
4. IN

1

213. Заокруживањем редног броја обележити клаузулу коју је потребно користи уколико листа иза резервисане речи SELECT садржи агрегатну функцију и једну или више колона које нису део агрегатне функције:

1. HAVING клаузулу
2. GROUP BY клаузулу
3. JOIN клаузулу
4. ORDER BY клаузулу

1

214. 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 објекта представља општим генеричким типом (надтипом).
3. Кардиналност неке везе одређује опционалност учешћа у вези.
4. Кардиналност показује колико кандидата за примарни кључ има неки тип ентитета.

2

215. Дата је табела PROJEKAT над којом се извршава упит:

```
ALTER TABLE PROJEKAT ADD RokKraj date
```

Оредити шта ће се десити након извршења упита. Заокружити број испред траженог одговора.

1. у табелу PROJEKAT додаје се ограничење RokKraj
2. у табелу PROJEKAT додаје се колона RokKraj
3. у табели PROJEKAT биће преименована колона
4. у базу података додаје се табела PROJEKAT са само једном колоном
5. у табели PROJEKAT промениће се тип података у колони RokKraj

2

216. Дата је табела **RADNIK** и упит:

IDBR	IME	PREZIME	PLATA	PREMIJA	DATZAP
6234	Marko	Pavlović	1300	3000	1990-12-17
6789	Janko	Nikolić	3900	10	1999-12-23

```
SELECT Ime, Prezime, DATEDIFF(year, DatZap, GETDATE()) AS God FROM Radnik
```

Одредити шта је резултат упита. Заокружити број испред траженог одговора:

2

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

217. 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	IMEOD	MESTO
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka
40	Nabavka	Barajevo

```
SELECT Imeod, AVG(Plata) AS ProspekPlata FROM Radnik, Odeljenje WHERE Odeljenje.Brod=Radnik.Brod GROUP BY Imeod HAVING AVG(Plata)>1000
```

Одредити резултат извршавања датог упита. Заокружити број испред траженог одговора:

2

1. Упит се не извршава зато што груписање мора да се изврши не само по називу одељења, него и по шифри одељења (BrOd)
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 називи одељења и висина просечне плате у њима, при чему се код одређивања просека узимају у обзир само плате веће од 1000

218. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Sloboda n	Golubović	900	10
5932	Mitar	Gavrilović	600	
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	IMEOD	MESTO
50	Skladišta	Zemun
30	Marketing	Vračar
10	Plasman	Surčin
20	Direkcija	Grocka

```
SELECT Odeljenje.Imeod, Radnik.Ime+' '+Radnik.Prezme as PunoIme
FROM Odeljenje LEFT JOIN Radnik ON Radnik.Brod = Odeljenje.Brod
```

Одредити шта се види као резултат датог упита. Заокружити број испред траженог одговора:

- Називи свих одељења – и оних у којима има радника и оних где нико није распоређен - са бројем радника у сваком одељењу
- За сваког распоређеног радника приказује се по један ред са називом одељења и пуним именом радника, док се за раднике који нису распоређени приказује само пуно име радника
- Приказује се по један ред за сваког распоређеног радника са називом одељења и пуним именом радника. За раднике који нису распоређени, као и за одељења у која нико није распоређен, не формирају се редови у резултујућој табели
- За свако одељење се приказује онолико редова колико радника ради у том одељењу, док се за одељења у којима нико не ради приказује по један ред са називом одељења

2

219. Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	BROD
5900	Sloboda n	Golubović	900	10
5932	Mitar	Gavrilović	600	10
5953	Persida	Kosanović	1100	20
6234	Marko	Pavlović	1300	30
6789	Janko	Nikolić	800	10

BROD	IMEOD	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
```

2

Одредити приказ који је резултат датог упита. Заокружити број испред траженог одговора:

- Бројева и назива свих одељења
- Бројева и назива свих одељења са бројем радника у њима
- Бројева и назива одељења у којима има радника са бројем радника у њима
- Бројева и назива одељења у којима нема радника

220. Извршава се следећа SELECT наредба:

```
SELECT MIN(Datum_Zaposlenja), Odsek_Id
FROM Zaposleni
GROUP BY Odsek_Id
```

Заокруживањем броја испред одговарајућег исказа, одредити које ће вредности бити приказане:

1. Најранији датум запослења за сваки одсек предузећа.
2. Најранији датум запослења у целом предузећу.
3. Датум запослења последњег запосленог радника у целом предузећу.
4. Датум запослења последњег запосленог радника за сваки одсек.
5. Датум запослења најстаријег запосленог радника у сваком одсеку предузећа.

2

221. 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 naziv
 FROM proizvod
 GROUP BY naziv
 HAVING cena > AVG(cena)
```
4. 

```
SELECT naziv
 FROM (SELECT AVG(cena) FROM proizvod)
 WHERE cena > AVG(cena)
```

2

222. 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:

1. Наредба се неће извршити једино ако нема ни једног одличног ученика.
2. Услов треба написати у HAVING клаузули
3. Потребно је назначити редослед сортирања (asc, desc).
4. Потребно је променити редослед клаузула.

2

223. Табела **ARTIKLI** садржи следеће колоне: *artikl\_id, naziv, kategorija, cena, kolicina*.  
Потребно је да се прикаже категорија и минимална цена артикла у свакој категорији.  
При томе се тражи приказ само оних категорија где је најмања цена производа већа од задате граничне вредности која се преноси упиту кроз параметар *@granica*

Изабрати упит који даје тражени извештај:

1. 

```
SELECT kategorija, MIN(cena)
 FROM artikli
 WHERE MIN(cena)>@granica
 GROUP BY cena
```
2. 

```
SELECT kategorija, MIN(cena)
 FROM artikli
 GROUP BY kategorija
 HAVING MIN(cena)>@granica
```
3. 

```
SELECT kategorija, MIN(cena)
 FROM artikli
 GROUP BY MIN(cena), kategorija
 HAVING MIN(cena)>@granica
```
4. 

```
SELECT kategorija, MIN(cena)
 FROM artikli
 WHERE MIN(cena)>@granica
 GROUP BY kategorija
```

2

224. Табела **RADIONICA** садржи следеће колоне: *radionica\_id, naziv, zanat, lokacija\_id*.  
Потребно је да се прикаже колико на свакој локацији има различитих заната.  
Заокружити број испред упита који даје тражени извештај:

1. 

```
SELECT DISTINCT location_id, COUNT(zanat)
 FROM radionica
 GROUP BY lokacija_id
```
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
```

2

225. Заокружити број испред одговора који представља наставак датог исказа:  
Уколико поглед (view) треба користити за измену података у табели, поглед **НЕ СМЕ**  
садржати...

1. WHERE клаузулу
2. Спој више табела
3. Алијас колоне
4. GROUP BY клаузулу

2

226. Дат је упит за креирање погледа и наведени искази који се односе на дати упит. Заокружити број испред тачног исказа:

```
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
```

1. Подаци у табели **Testovi** се могу модификовати коришћењем погледа **Pregled\_Proseka**
2. Коришћењем датог погледа, подаци се могу само у додавати у табелу **Testovi**, али не и мењати
3. Овако дат упит изазива грешку при извршењу
4. Коришћењем датог погледа, подаци из табеле **Testovi** се могу само прегледавати, али не и додавати или мењати

2

### У следећим задацима заокружите бројеве испред тражених одговора

- 227 Заокружити бројеве испред тражених одговора. Међу понуђеним алатима, обележити **Case** алате:

1. Rational Rose
2. Oracle Designer
3. .NET
4. Microsoft Visio
5. Java
6. SQL Express

1,5

- 228 Обележити команде које се сматрају командама ажурирања података у бази података. Заокружити бројеве испред тражених одговора:

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

1,5

- 229 Заокружити бројеве испред тражених одговора.

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 при чему се добија растући поредак
6. службена реч се може изоставити при чему се добија опадајући поредак

1,5

230	Заокружити бројеве испред наредби које служе за креирање, брисање и измену структуре релационе базе и објеката који чине релациону базу:	1,5
	1. ALTER TABLE 2. INSERT 3. CREATE TABLE 4. DROP TABLE 5. UPDATE 6. DELETE TABLE 7. ADD COLUMN 8. ADD CONSTRAINT	
231	Заокружити бројеве испред тражених одговора.  Одредити ентитетете који садрже одговарајуће атрибуте:	2
	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	
232	Дати су искази који се односе на спољашњи кључ табеле ( <i>foreign key constraint</i> ). Заокружити бројеве испред <b>тачних</b> исказа:	2
	1. Вредност у пољу спољашњег кључа не сме бити NULL 2. Вредност спољашњег кључа мора бити јединствена ( <i>unique</i> ) у колони над којом је постављено ограничење спољашњег кључа 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	
233	Заокружити бројеве испред тражених одговора.  Означити операторе који се <b>НЕ МОГУ</b> користити за поређење са подупитом који враћа више вредности:	2
	1. ALL 2. ANY 3. BETWEEN 4. SOME 5. LIKE 6. IN	

234 Дата је табела **Kupci** са структуром:

```
(Id int primary key, Prezime varchar(50), Adresa varchar(50), Mesto varchar(20), Telefon varchar(5), Status varchar(8))
```

И табела **NoviKupci** са структуром:

```
(Id int primary key, Prezime varchar(50), Telefon varchar(20), 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. Како табела **NoviKupci** има све колоне које постоје и у табели **Kupci**, упит се извршава без грешке и у табелу **NoviKupci** се уписују записи из табеле **Kupci** са статусом који није **Aktivan**
2. Упит се не извршава, пријављује грешку јер се број колона у табели **Kupci** разликује од броја колона у табели **NoviKupci**
3. Упит би се извршио без грешке да су у SELECT клаузули подупита, уместо \* наведене све колоне табеле **Kupci** које имају своју одговарајућу колону у табели **NoviKupci**
4. Упит јавља грешку због покушаја уписа вредности у поље примарног кључка који је аутоматски, тј креира га сама база

235 Дата је табела **RADNIK**, табела **ODELJENJE** и упит:

IDBR	IME	PREZIME	PLATA	SIFRAOD
5900	Slobodan	Golubović	900	10
5932	Mitar	Gavrilović	600	
5953	Persida	Kosanović	1100	20

SIFRAOD	IMEOD	MESTO
10	Marketin g	Vračar
20	Direkcija	Grocka
30	Nabavka	Barajev o

Креирана је ускладиштена процедура са параметром @br int = NULL

Позивом процедуре извршава се упит:

```
UPDATE Radnik SET Radnik.SifraOD = 30
WHERE Radnik.SifraOD=@br or @br IS NULL
```

2

Одредити који од понуђених исказа су тачни. Заокружити бројеве испред тражених одговора:

1. Сви радници који раде у одељењу са шифром једнакој вредности која је пренета кроз параметар @br, биће прераспоређени у одељење чија је шифра 30
2. Упит распоређује све нераспоређене раднике у одељење са шифром 30
3. Уколико се параметру @br не пренесе вредност, радници који су до тог момента били нераспоређени, биће распоређени у одељење са шифром 30
4. Уколико се параметру @br не пренесе вредност, СВИ радници ће бити прераспоређени у одељење чији је број 30

236 Табела **Zaposleni** садржи поља: Zaposleni\_Id, Ime, Prezime, Plata, Odsek\_Id.

Дат је упит:

```
SELECT Zaposleni_Id, Ime, Prezime
FROM Zaposleni
WHERE Plata=(SELECT MAX(Plata) FROM Zaposleni GROUP BY Odsek_Id)
```

Дати су искази који описују ефекат извршења упита. Заокружити бројеве испред ТАЧНИХ исказа:

1. Упит се не извршава зато што у подупиту није дозвољено коришћење групних функција
2. Упит се извршава без грешке и из сваког одељења бира и приказује податке о раднику који има највећу плату у том одељењу.
3. Упит се не извршава јер подупит враћа више од једне врсте, а коришћен је оператор за поређење са једном вредношћу.
4. Уколико би се изоставила GROUP BY клаузула, упит би се извршавао без грешке и приказао би једног или више радника са платом једнакој највећој плати (без обзира на одељење).
5. Како подупит садржи груписање, да би се упит извршио без грешке, потребно је услов са подупитом написати у HAVING уместо у WHERE клаузули

2

237 Заокружити бројеве под којима су наведене клаузуле SQL наредбе у којима се могу користити аритметичке операције:

1. SELECT
2. FROM
3. WHERE
4. ORDER BY

2

238 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

239 Заокружити бројеве испред команди које се могу користити за ажурирање постојећих података у бази:

1. DELETE
2. MERGE
3. SELECT
4. UPDATE

2

- 240 Креирање су табеле **SKOLA** и **OSNOVNAŠKOLA**, а затим су у табелу **SKOLA** уписани подаци извршавањем следећих наредби:

```
create table Skola(
 skolaID int primary key, Naziv varchar(50), gradID int, tip
 varchar(50))
create table OsnovnaSkola(
 gimID int primary key, Naziv varchar(50), gradID int)

insert into Skola values (101,'Nikola Tesla',20, 'srednja strucna')
insert into Skola values (102,'Dusko Radovic', 20, 'osnovna')
insert into Skola values (103,'Sveti Sava', 30, 'osnovna')
insert into Skola values (104,'Bora Stankovic', 20, 'gimnazija')
```

У табелу **OSNOVNAŠKOLA** треба уписати податке о основним школама преписивањем потребних вредности из табеле **SKOLA**. Заокружити бројеве испред упита који ће јавити грешку при извршењу:

1. `select * into OsnovneSkole  
from Skola where tip='osnovna'`
2. `insert into OsnovneSkole  
select * from Skola where tip='osnovna'`
3. `insert into OsnovneSkole(skolaID, Naziv)  
select s.skolaID, s.Naziv from Skola as s where tip='osnovna'`
4. `insert into OsnovneSkole  
select s.skolaID, s.Naziv, s.gradID from Skola as s where  
tip='osnovna'`

- 241 Креирање су и попуњене подацима табеле **Korisnik** и **Prijatelji**. Њихова структура и садржај приказани су на слици:

**Korisnik**

ID	IME	POL
1	Ana	NULL
2	Steva	m
3	Marta	z
4	Petra	z

**Prijatelji**

Korisnik1	Korisnik2
1	2
1	3
2	3

Извршавањем упита добија се табела са подацима.

```
select k.ime, COUNT(*) as [broj prijatelja]
from Korisnik as k
left join Prijatelji as p on p.korisnik1=k.id or p.korisnik2=k.id
where k.pol='z'
group by k.id, k.ime
```

Заокруживањем бројева испред понуђених одговора, обележити који од наведених података ће бити приказани у појединим редовима резултујуће табеле:

- |             |             |
|-------------|-------------|
| 1. Ana, 1   | 5. Marta, 1 |
| 2. Ana, 2   | 6. Marta, 2 |
| 3. Steva, 1 | 7. Petra, 0 |
| 4. Steva, 2 | 8. Petra, 1 |

2

2

242 Заокружити бројеве испред кључних речи које се НЕ КОРИСТЕ за обележавање ограничења (*constraints*) у језику SQL:

1. Foreign key
2. Unique
3. Distinct
4. Check
5. Convert
6. Union
7. Not Null
8. Except

2

243 Одредити ентитете који садрже одговарајуће атрибуте. Заокружити бројеве испред тражених одговора:

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

244 Обележити ентитете код којих је извршен адекватан избор јединственог идентификатора. Заокружи бројеве испред тражених одговора:

1. јединствени матични број грађанина (ЈМБГ) за ентитет ОСОБА
2. датум рођења за ентитет ОСОБА
3. ИСБН број за ентитет КЊИГА
4. регистарска ознака за АУТОМОБИЛ
5. дестинација за ентитет АРАНЖМАН
6. режисер за ентитет ФИЛМ

3

245 Одредити тачан исказ о оператору ANY који се примењује са подупитом који враћа више вредности:

1. Оператор ANY може да се користи испред кључне речи DISTINCT.
2. Оператор ANY врши поређење са свим вредностима које враћа подупит и враћа TRUE ако све вредности подупита задовољавају услов
3. Оператор ANY врши поређење са свим вредностима које враћа подупит и враћа TRUE ако било која од вредности подупита задовољава услов
4. Оператору ANY може да претходи оператор LIKE или оператор IN.
5. Оператору ANY мора да претходи оператор поређења (=, <>, >, >=, <, <=)
6. Услов =ANY(скуп вредности) је еквивалентан услову IN (скуп вредности)

3

- 246 Креирана је табела **SKOLA**, а затим су у њу уписани подаци извршавањем следећих наредби:

```
create table Skola(skolaID int primary key, Naziv varchar(50))
insert into Skola values (101,'Nikola Tesla')
insert into Skola values (102,'Mihajlo Pupin')
insert into Skola values (103,'ETS Zemun')
```

За дати упит, треба проценити сценарио који ће се десити и заокруживањем редних бројева испред исказа, означити могуће исходе:

```
select * into StrucneSkole from Skola
```

1. Креира се копија табеле Skola - нова табела под именом StrucneSkole исте структуре као и табела Skola и у њу се преписују сви подаци из табеле Skola
2. Уколико табела са именом StrucneSkole постоји у бази, креира се нова са именом StrucneSkole(1) и у њу се преписују сви редови из табеле Skola
3. Уколико табела са именом StrucneSkole постоји у бази, не креира се нова, само се у постојећу преписују редови из табеле Skola
4. Уколико табела са именом StrucneSkole постоји у бази, упит јавља грешку
5. Уколико се дода услов `where 1=2` упит се извршава, креира се нова табела исте структуре као и табела Skola, али се у њу не уписује ни један ред
6. Уколико се дода услов `where 1=2` упит јавља грешку јер је `1=2` увек нетачно тј. **False**
7. Упит јавља грешку јер се кључна реч `into` користи искључиво у комбинацији са `insert`

### Допуните следеће реченице и табеле

247. Допунити реченицу наводећи назив нормалне форме:

Уколико ни један атрибут релације није вишевредносни, нити композитни, тј. не може се раставити, кажемо да је релација у \_\_\_\_\_ нормалној форми.

1

248. Допунити реченицу наводећи назив нормалне forme:

Уколико сви атрибути релације који нису део кључа зависе од сваког атрибута који је део кључа кажемо да је релација у \_\_\_\_\_ нормалној форми.

1

249. Допунити реченицу наводећи назив нормалне forme:

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

## У следећим задацима уредите и повежите појмове према захтеву

250. Написати на цртама испред логичких операција редне бројеве њихових приоритета:

1. највиши приоритет \_\_\_\_\_ OR

2. средњи приоритет \_\_\_\_\_ NOT

3. најнижи приоритет \_\_\_\_\_ AND

1,5

251. Дата је табела **GEOGRAFIJA** која поред осталих података садржи називе градова и држава (*Naziv nvarchar(50)*). У зависности од услова у **WHERE** клаузули, **SELECT** упитом се приказују географски појмови из табеле. Са леве стране су дати услови нумерисани бројевима од 1 до 5, а са десне групе градова.

Свакој групи градова придржити по један услов уносом редног броја коим је услов нумерисан на линију испред листе градова:

1. `where Naziv like 'L_ %'` \_\_\_\_\_ SIJERA LEONE, SVETA LUCIJA

2,5

2. `where Naziv like '_ %N%'` \_\_\_\_\_ LA VALETA, LA KORUNJA

3. `where Naziv like '% L%'` \_\_\_\_\_ EL RENO, LA KORUNJA

4. `where Naziv like '_ LX'` \_\_\_\_\_ EL SALVADOR, EL RENO

5. `where Naziv like '_ %A'` \_\_\_\_\_ LAS VEGAS, LOS ANGELES

252. Исписати на цртама испред релација редни број под којим је наведена одговарајућа кардиналност везе:

1. 1 : 1 \_\_\_\_\_ ВЛАСНИК – БРОЈ ТЕЛЕФОНА

2,5

2. 1 : M \_\_\_\_\_ НАСТАВНИК – ПРЕДМЕТ

3. M : M \_\_\_\_\_ ОСОБА – ПАСОШ

\_\_\_\_\_ КУПАЦ – МОДЕЛ АУТОМОБИЛА

\_\_\_\_\_ УТАКМИЦА – ГРАД ДОМАЋИН

253. Уписати редни број почев од 1 на линију испред резервисане речи тако да одговара редоследу навођења при формирању упита.

За формирање упита за издавање дела података из табеле која се налази у оквиру базе података користе се клаузуле у следећем редоследу:

\_\_\_\_\_ GROUP BY

3

\_\_\_\_\_ WHERE

\_\_\_\_\_ SELECT

\_\_\_\_\_ ORDER BY

\_\_\_\_\_ FROM

254. 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

BR OD	IMEOD	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

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

255. 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	IMEOD	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

`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

3

3

256. На левој страни су наведене категорије SQL команди, а са десне су набројане команде. На линију испред команда уписати број под којим је наведена категорија којој команда припада:

- |                                       |              |
|---------------------------------------|--------------|
| 1. DDL – Data Definition Language     | _____ GRANT  |
| 2. DML – Data Manipulation Language   | _____ UPDATE |
| 3. DCL – Data Control Language        | _____ COMMIT |
| 4. TCL – Transaction Control Language | _____ DROP   |
|                                       | _____ DELETE |
|                                       | _____ ALTER  |

3

257. Табела **Zaposleni** је креирана и попуњена извршавањем следећих наредби:

```
create TABLE zaposleni(
 id INTEGER NOT NULL PRIMARY KEY, rukovodilacId INTEGER, ime VARCHAR(30)
NOT NULL,
 FOREIGN KEY (rukovodilacId) REFERENCES zaposleni(id))

INSERT INTO zaposleni VALUES(1, NULL, 'Petar');
INSERT INTO zaposleni VALUES(2, 1, 'Mihajlo');
INSERT INTO zaposleni VALUES(3, 2, 'Milica');
INSERT INTO zaposleni VALUES(4, 3, 'Lazar');
INSERT INTO zaposleni VALUES(5, NULL, 'Sofija');
```

Очекивани ефекти извршења упита нумерисани су бројевима од 1 до 6. Уносом редног броја одговарајућег описа на предвиђену линију испред упита, повезати упит и опис резултата његовог извршења:

- |                                                         |         |
|---------------------------------------------------------|---------|
| 1. Сви радници који су руководиоци неком другом раднику | _____ 3 |
| 2. Сви радници који нису руководиоци ником              | _____   |
| 3. Сви радници који немају надређене руководиоце        | _____   |
| 4. Сви радници који имају надређеног руководиоца        | _____   |
| 5. Празна табела                                        | _____   |

```
_____ select * from zaposleni
_____ where id not in (select distinct rukovodilacId from zaposleni)

_____ select * from zaposleni
_____ where id not in
_____ (select rukovodilacId from zaposleni where rukovodilacId is not null)

_____ select * from zaposleni where rukovodilacId is null
```

3

### **АНЕКС 3. РАДНИ ЗАДАЦИ СА ОБРАСЦИМА ЗА ОЦЕЊИВАЊЕ**

РАДНА ВЕРЗИЈА

Поштовани ученици, ментори и оцењивачи,

Пред вами су документи који садрже радне задатаке и обрасце за оцењивање који ће бити заступљени на матурском практичном раду за образовни профил електротехничар информационих технологија. Намењени су за вежбање и припрему за полагање матурског испита, као и оцењивачима за усвајање примењене методологије оцењивања.

Задаци су рађени према компетенцијама: *Припрема и организација рада, Израда десктоп апликација, Израда веб апликација (статичких и динамичких веб страница), Израда и управљање базом података* и *Израда техничке документације* које се проверавају на испиту. У оквиру два сложена радна задатка обједињени су захтеви свих делова, јединица компетенција наведних у стандарду квалификације. У оквиру задатака проверава се ученикова компетентност и у погледу ефикасног планирања организације времена, савесног, одговорног, уредног, прецизног обављања послова, испољавања аналитичке способности при раду, као и испољавања позитиваног односа према значају спровођења прописа и важећих стандарда у електротехници и рачунарству, и испољавања позитиваног односа према професионално - етичким нормама и вредностима.

Задатком је предвиђено да се ученик постави у професионалну ситуацију док извршава послове **Електротехничар информационих технологија**.

Сваки радни задатак доноси **максимално 100 бодова**. Ученик мора остварити **најмање 50 бодова на сваком практичном задатку** како би положио испит. Обрасци за оцењивање садрже утврђене аспекте, индикаторе оцењивања као и одговарајуће мере процене дате кроз двостепену скалу.

Правилно обављање операција приликом практичног извођења подразумева да је ученик способан да **самостално** обавља радне задатке, показује да поседује неопходна знања и вештине за извршавање комплексних послова и повезивање различитих корака у оквиру њих; преузима одговорност за примену процедуре, средстава и организацију сопственог рада. Сви наведени критеријуми морају бити узети у обзир приликом процене компетентности.

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

## ЛИСТА РАДНИХ ЗАДАКА

По завршеном образовању за образовни профил **Електротехничар информационих технологија** ученик стиче стручне компетенције које се проверавају одговарајућим радним задацима како је то наведено у следећој табели.

Стручна компетенција	Шифра радног задатка	Назив радног задатка
A  ИЗРАДА ДЕСКТОП АПЛИКАЦИЈА, ПРОЈЕКТОВАЊЕ ЈЕДНОСТАВНЕ БАЗЕ ПОДАТКА И ПОВЕЗИВАЊЕ АПЛИКАЦИЈЕ СА БАЗОМ ПОДАТКА	EIT-A01	Библиотека
	EIT-A02	Библиотека
	EIT-A03	Евиденција радника
	EIT-A04	Сеоски туризам
	EIT-A05	Дневни боравак деце
	EIT-A06	Половни аутомобили
	EIT-A07	Факултетска евиденција
	EIT-A08	Археолошка налазишта
	EIT-A09	Фудбалски клубови
	EIT-A10	Риболовачко друштво
	EIT-A11	Евиоденција књига
	EIT-A12	Фудбалски стадиони
	EIT-A13	ДВД колекција
	EIT-A14	Дистрибуција лекова
	EIT-A15	Клинички центар
	EIT-A16	Изложба паса
	EIT-A17	Евиденција возила
	EIT-A18	Авто плац „Гараж“
	EIT-A19	Евиденција запослених
	EIT-A20	Туристичка агенција
	EIT-A21	Туристички обиласци
	EIT-A22	Голф клуб
	EIT-A23	Позоришне представе
	EIT-A24	Поликлиника
	EIT-A25	Заказивање прегелда
	EIT-A26	Хотелске резервације
	EIT-A27	Осигурање возила
	EIT-A28	Стовариште материјала
	EIT-A29	Голф играчи
	EIT-A30	Голф терени

Стручна компетенција	Шифра радног задатка	Назив радног задатка
Б ИЗРАДА ВЕБ СТРАНИЦЕ И ВЕБ АПЛИКАЦИЈЕ И ОБЈАВА НА ВЕБ СЕРВЕРУ	ЕИТ-Б01	Фото галерија
	ЕИТ-Б02	Домаће животиње
	ЕИТ-Б03	Србија и њени суседи
	ЕИТ-Б04	Слајдер за слике
	ЕИТ-Б05	Звуци инструмената
	ЕИТ-Б06	Успех ученика
	ЕИТ-Б07	Библиотека
	ЕИТ-Б08	Књига утисака
	ЕИТ-Б09	ЦД каталог
	ЕИТ-Б10	Електронски речник
	ЕИТ-Б11	Временска прогноза
	ЕИТ-Б12	Резервација аутобуских карата
	ЕИТ-Б13	Телефонски именик
	ЕИТ-Б14	ТВ програм
	ЕИТ-Б15	Ред вожње
	ЕИТ-Б16	Распоред часова
	ЕИТ-Б17	Веб продавница
	ЕИТ-Б18	Временска прогноза
	ЕИТ-Б19	Конвертор текста
	ЕИТ-Б20	Курсна листа
	ЕИТ-Б21	Сортирање табеле
	ЕИТ-Б22	Ефекти табеле

### ПРИПРЕМА КОМБИНАЦИЈА РАДНИХ ЗАДАТКА ЗА МАТУРСКИ ИСПИТ

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.

Комбинацију на матурском испиту чини један задатак из групе А и један задатак из групе Б. Све потенцијалне комбинације су могуће, односно сваки задатак из групе А се може комбиновати са сваким задатком из групе Б, уз услов да се, при припреми комбинација за испитни рок, један задатак не може поновити више од 4 пута.

Листа комбинација дата је у табели ниже.

## КОМБИНАЦИЈЕ РАДНИХ ЗАДАТКА ЗА МАТУРСКИ ПРАКТИЧНИ РАД

комбинација број	радни задаци	комбинација број	радни задаци	комбинација број	радни задаци
1	ЕИТ-А01 ЕИТ-Б01	2	ЕИТ-А01 ЕИТ-Б02	3	ЕИТ-А01 ЕИТ-Б03
4	ЕИТ-А01 ЕИТ-Б04	5	ЕИТ-А01 ЕИТ-Б05	6	ЕИТ-А01 ЕИТ-Б06
7	ЕИТ-А01 ЕИТ-Б07	8	ЕИТ-А01 ЕИТ-Б08	9	ЕИТ-А01 ЕИТ-Б09
10	ЕИТ-А01 ЕИТ-Б10	11	ЕИТ-А01 ЕИТ-Б11	12	ЕИТ-А02 ЕИТ-Б12
13	ЕИТ-А02 ЕИТ-Б13	14	ЕИТ-А02 ЕИТ-Б14	15	ЕИТ-А02 ЕИТ-Б15
16	ЕИТ-А02 ЕИТ-Б16	17	ЕИТ-А02 ЕИТ-Б17	18	ЕИТ-А02 ЕИТ-Б18
19	ЕИТ-А02 ЕИТ-Б19	20	ЕИТ-А02 ЕИТ-Б20	21	ЕИТ-А02 ЕИТ-Б21
22	ЕИТ-А02 ЕИТ-Б22	23	ЕИТ-А03 ЕИТ-Б01	24	ЕИТ-А03 ЕИТ-Б02
25	ЕИТ-А03 ЕИТ-Б03	26	ЕИТ-А03 ЕИТ-Б04	27	ЕИТ-А03 ЕИТ-Б05
28	ЕИТ-А03 ЕИТ-Б06	29	ЕИТ-А3 ЕИТ-Б07	30	ЕИТ-А03 ЕИТ-Б08
31	ЕИТ-А03 ЕИТ-Б09	32	ЕИТ-А03 ЕИТ-Б10	33	ЕИТ-А03 ЕИТ-Б11
34	ЕИТ-А04 ЕИТ-Б12	35	ЕИТ-А04 ЕИТ-Б13	36	ЕИТ-А04 ЕИТ-Б14
37	ЕИТ-А04 ЕИТ-Б15	38	ЕИТ-А04 ЕИТ-Б16	39	ЕИТ-А04 ЕИТ-Б17
40	ЕИТ-А04 ЕИТ-Б18	41	ЕИТ-А04 ЕИТ-Б19	42	ЕИТ-А04 ЕИТ-Б20
43	ЕИТ-А04 ЕИТ-Б21	44	ЕИТ-А04 ЕИТ-Б22	45	ЕИТ-А05 ЕИТ-Б01
46	ЕИТ-А05 ЕИТ-Б02	47	ЕИТ-А05 ЕИТ-Б03	48	ЕИТ-А05 ЕИТ-Б04
49	ЕИТ-А05 ЕИТ-Б05	50	ЕИТ-А05 ЕИТ-Б06	51	ЕИТ-А05 ЕИТ-Б07
52	ЕИТ-А05 ЕИТ-Б08	53	ЕИТ-А05 ЕИТ-Б09	54	ЕИТ-А05 ЕИТ-Б10
55	ЕИТ-А05 ЕИТ-Б11	56	ЕИТ-А06 ЕИТ-Б12	57	ЕИТ-А06 ЕИТ-Б13
58	ЕИТ-А06 ЕИТ-Б14	59	ЕИТ-А06 ЕИТ-Б16	60	ЕИТ-А06 ЕИТ-Б16
61	ЕИТ-А06 ЕИТ-Б17	62	ЕИТ-А06 ЕИТ-Б18	63	ЕИТ-А06 ЕИТ-Б19
64	ЕИТ-А06 ЕИТ-Б20	65	ЕИТ-А06 ЕИТ-Б21	66	ЕИТ-А06 ЕИТ-Б22
67	ЕИТ-А07 ЕИТ-Б01	68	ЕИТ-А07 ЕИТ-Б02	69	ЕИТ-А07 ЕИТ-Б03
70	ЕИТ-А07 ЕИТ-Б04	71	ЕИТ-А07 ЕИТ-Б05	72	ЕИТ-А07 ЕИТ-Б06
73	ЕИТ-А07 ЕИТ-Б07	74	ЕИТ-А07 ЕИТ-Б08	75	ЕИТ-А07 ЕИТ-Б09
76	ЕИТ-А07 ЕИТ-Б10	77	ЕИТ-А07 ЕИТ-Б11	78	ЕИТ-А08 ЕИТ-Б12

<b>79</b>	ЕИТ-А08 ЕИТ-Б13	<b>80</b>	ЕИТ-А08 ЕИТ-Б14	<b>81</b>	ЕИТ-А08 ЕИТ-Б15
<b>82</b>	ЕИТ-А08 ЕИТ-Б16	<b>83</b>	ЕИТ-А08 ЕИТ-Б17	<b>84</b>	ЕИТ-А08 ЕИТ-Б18
<b>85</b>	ЕИТ-А08 ЕИТ-Б19	<b>86</b>	ЕИТ-А08 ЕИТ-Б20	<b>87</b>	ЕИТ-А08 ЕИТ-Б21
<b>88</b>	ЕИТ-А08 ЕИТ-Б22	<b>89</b>	ЕИТ-А09 ЕИТ-Б01	<b>90</b>	ЕИТ-А09 ЕИТ-Б02
<b>91</b>	ЕИТ-А09 ЕИТ-Б03	<b>92</b>	ЕИТ-А09 ЕИТ-Б04	<b>93</b>	ЕИТ-А09 ЕИТ-Б05
<b>94</b>	ЕИТ-А09 ЕИТ-Б06	<b>95</b>	ЕИТ-А09 ЕИТ-Б07	<b>96</b>	ЕИТ-А09 ЕИТ-Б08
<b>97</b>	ЕИТ-А09 ЕИТ-Б09	<b>98</b>	ЕИТ-А09 ЕИТ-Б10	<b>99</b>	ЕИТ-А09 ЕИТ-Б11
<b>100</b>	ЕИТ-А10 ЕИТ-Б12	<b>101</b>	ЕИТ-А10 ЕИТ-Б13	<b>102</b>	ЕИТ-А10 ЕИТ-Б14
<b>103</b>	ЕИТ-А10 ЕИТ-Б15	<b>104</b>	ЕИТ-А10 ЕИТ-Б16	<b>105</b>	ЕИТ-А10 ЕИТ-Б17
<b>105</b>	ЕИТ-А10 ЕИТ-Б18	<b>107</b>	ЕИТ-А10 ЕИТ-Б19	<b>108</b>	ЕИТ-А10 ЕИТ-Б20
<b>109</b>	ЕИТ-А10 ЕИТ-Б21	<b>110</b>	ЕИТ-А10 ЕИТ-Б22	<b>111</b>	ЕИТ-А11 ЕИТ-Б01

У Табели је наведена *само 101 комбинација* од могућих 660 комбинација радних задатака. Све потенцијалне комбинације су могуће на матурском раду. Листу радних задатака, обрасце за оцењивање радних задатака, и листу комбинација, Центар доставља школама у оквиру овог Приручника.

## РАДНИ ЗАДАЦИ - КОМПЕТЕНЦИЈА А

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А1

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Библиотека**

Управа градске библиотеке је исказала потребу за креирањем апликације за вођење евиденције о насловима који су на располагању у библиотеци. Апликација би требала осим могућности уноса, брисања и измена података о расположивим насловима, писцима, корисницима и изнајмљивању да има и могућност генерирања потребних извештаја. Део активности у изради апликације делегиран је вашем тиму.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

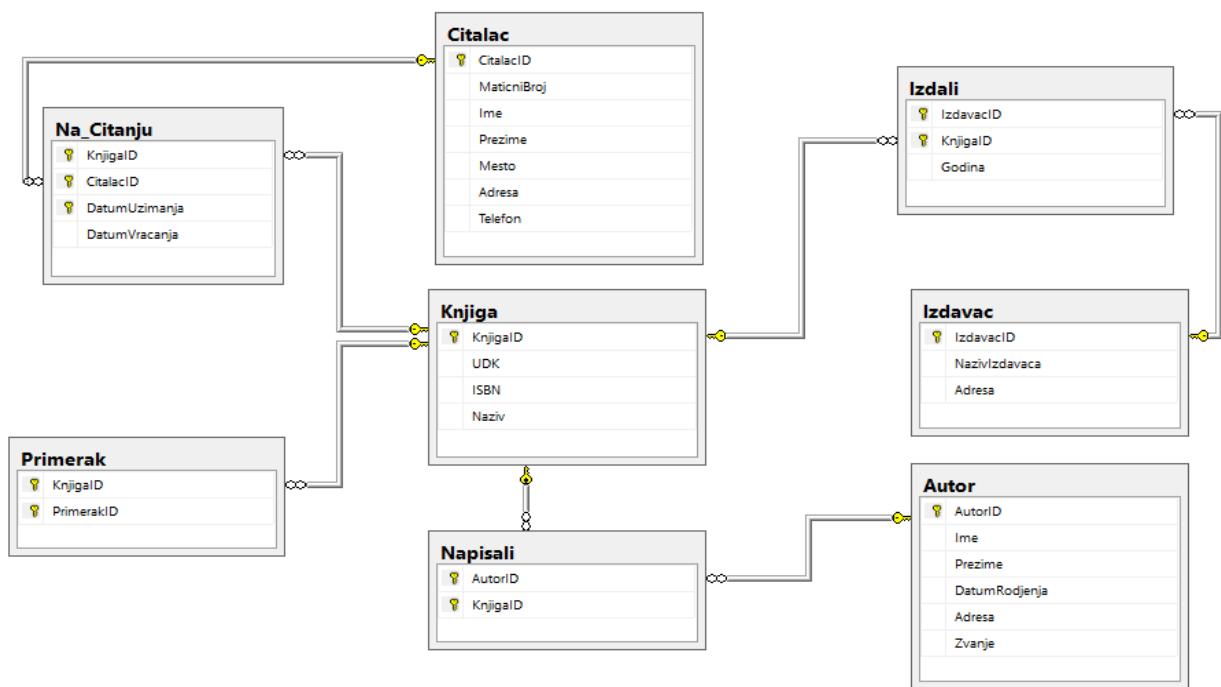
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A11**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A12**.

Архива слика и икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:

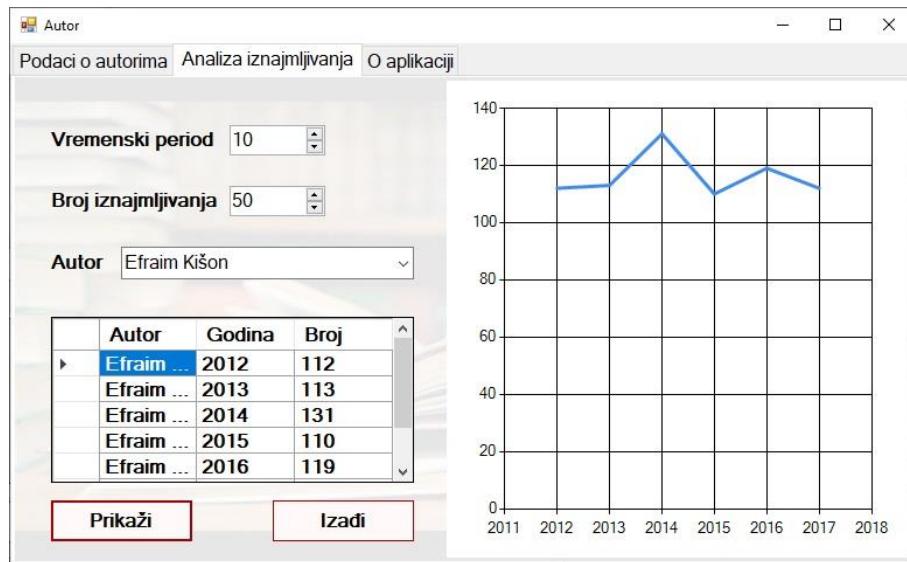


Креирати форму која има три таба. На првом табу се приказују подаци о свим ауторима чијим делима библиотека располаже. Податке о свим ауторима приказује у **listView** контроли.

Уносом шифре у одговарајуће поље, уколико постоји аутор са унетом шифром, податке о том аутору треба приказати у осталим пољима форме. Ако унета шифра не постоји у бази или ако је поље шифре празно, ресетовати садржај осталих текстуалних поља на форми.

Šifra	Ime	Prezime	Datum rođenja	Adres
1	Efraim	Kišon	13.08.1980	Louisian
2	Džerom	Džerom	24.11.1973	South C
3	Ivo	Andrić	14.02.1985	Massac
4	Aleksa	Šantić	15.05.1991	New Ha
5	Jovan	Dučić	26.01.1978	Utah, S
6	Branko	Čopić	06.03.1972	Nevada,
7	Branko	Radičević	22.02.1969	South D
8	Branisl...	Nušić	16.10.1959	Virginia,
9	Lav Ni...	Tolstoj	24.03.1991	Californi
10	Dostojev...	Dostojev...	05.10.1966	Marylan
	Safon	s...	09.06.1957	Ohio, Wi
	Braun	...	07.12.1962	Idaho, L
	Ljosa	...	28.07.2000	Alabama
	Ruljnakov	...	09.07.1997	North D

Кликом на дугме **Упиши аутора**, уписати аутора чији су подаци унети у поља форме и уколико је упис успешан, приказати шифру на коју је у бази уписан нови аутор. После успешног уноса променити приказ на форми. Да не би случајним кликом на дугме за упис дошло до вишеструког уписивања истог аутора, дугме учинити доступним (енејбловано) само онда када су сва поља за унос података о новом аутору празна. После уноса, дугме **Упиши** треба поново искључити (дисејбловати). Дугме **Изађи** затвара целу апликацију.



Други таб форме организовати као на скици. **ComboBox** треба попунити именима и презименима аутора (сортирано у алфавитном поретку). Кликом на дугме **Прикажи**, потребно је за избраног аутора, током задатог временског периода израженог у годинама пребројати колико су пута књиге избраног аутора изнајмљене од стране корисника библиотеке (нпр током протеклих 10 година).

Приказују се пуно име аутора, календарска година и број књига који је издат током наведене године. При томе нас не интересују подаци из свих година, већ само подаци из оних током којих су дела означеног аутора издата више од неког специфицираног броја пута. Податке приказати на форми у **dataGridView** контроли, као и на графику. На x оси су године, а на y оси број изнајмљивања.

На трећем табу је исписано кратко корисничко упутство и документација о апликацији.

Библиотека отвара већи број огранака у школама у околини и потребно је чувати податке и о школама у којима су отворени огранци библиотеке. За сваки примерак књиге треба памтити у ком огранку се налази Такође је потребно памтити и када је књига набављена. Обезбедити кроз ограничења у бази да датум набавке не може да буде старији од датума оснивања библиотеке (01.01.2000). Креирати дијаграм и бекап стања проширене базе.

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А2**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Библиотека**

Пројектована је база за памћење и обраду података о расположивим ресурсима библиотеке и њеним корисницима. Структура базе је дата на слици у прилогу. Управа библиотеке се труди да стално набавља нове наслове, али и расходује старе – похабане или изгубљене наслове. Да не би дошло до претрпавања података, тражено је да се омогући преглед и брисање аутора (у ситуацији када библиотека не поседује ни један наслов тог аутора). Исказана је потреба да се у бази чувају и додатни подаци о формату у коме је књига штампана.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

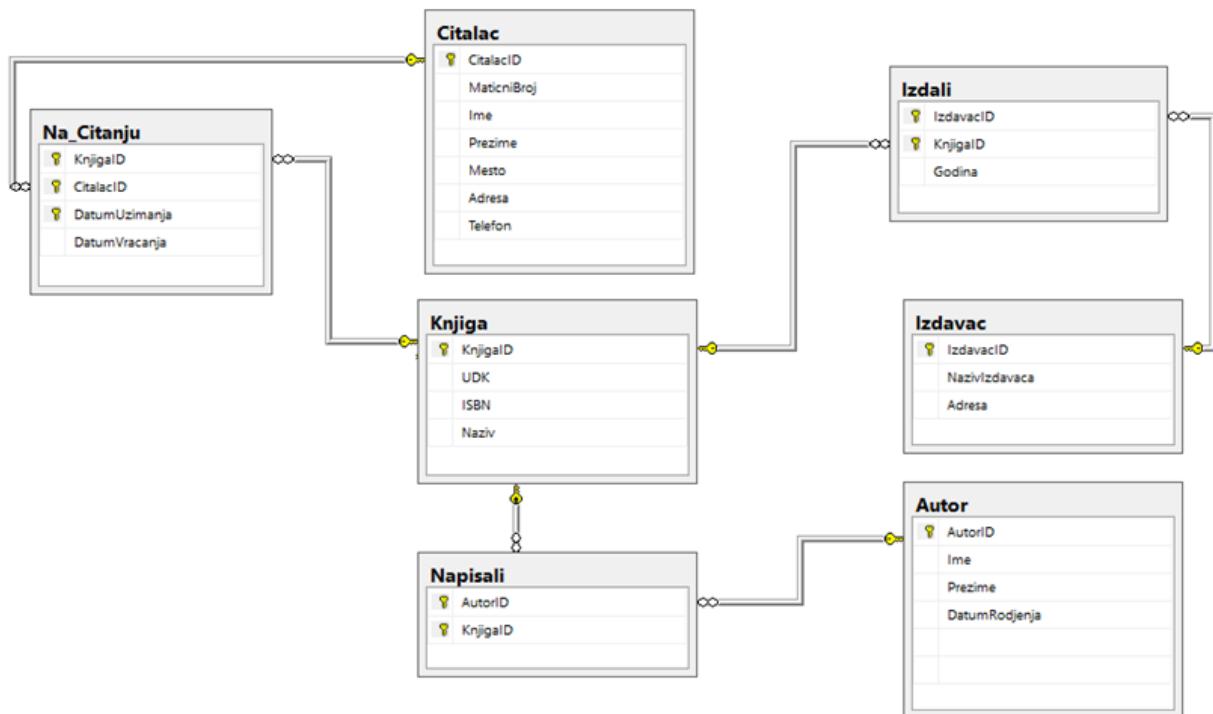
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A21**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A22**.

Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која податке о свим ауторима табеларно приказује у **listView** контроли.

Кликом на неки од редова, податке о аутору треба приказати у осталим пољима форме.

Кликом на дугме **Брисање** у линији са алатима, обрисати изабраног аутора из базе, а потом освежити приказ у **listView-у**.

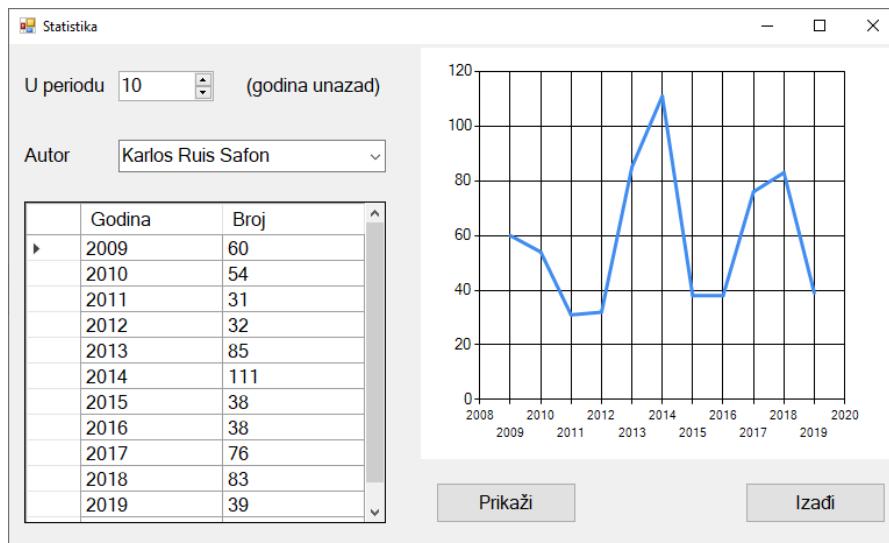
Кликом на дугме **Излаз** у линији алата прве форме, затворити цеу апликацију.

Отворити кратко корисничко упутство кликом на дугме **О апликацији**.

Šifra	Ime	Prezime	Datum rođenja
1	Efraim	Kišon	13/08/1980
2	Džerom	Džerom	24/11/1973
3	Ivo	Andrić	14/02/1985
4	Aleksa	Šantić	15/05/1991
5	Jovan	Dučić	26/01/1978
6	Branko	Ćopić	06/03/1972
7	Branko	Radičević	22/02/1969
8	Branislav	Nušić	16/10/1959
9	Lav Nikolaj...	Tolstoj	24/03/1991
10	Fjodor M.	Dostoevski	05/10/1966
11	Karlos Ruis	Safon	09/06/1957

Дугме **Анализа** отвара другу форму на којој је потребно приказати колико су пута дела изабраног аутора била изнајмљена током задатог временског периода. Временски период се задаје у односу на текући датум, као изабрани број година уназад. Табела и график се приказују кликом на дугме **Прикажи**.

Кликом на **Изађи** вратити корисника на почетну форму.



Осмислите проширење базе којим се за сваки примерак књиге омогућава памћење формата у коме је књига издана. Format може да буде тврди повез, меки повез, илустровано издање...

Проширити базу тако да се за читаоца памтити датум последњег плаћања чланарине и ограничiti да се чланарина не може да плаћа ретроактивно (датум плаћања није старији од текућег датума).

Креирати дијаграм и бекап стања проширене базе.

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А3**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Евиденција радника**

Клијент чува податаке о изведеним пројектима и упосленицима на пројектима у бази података. Структура базе је дата на слици у прилогу. Велики број одрађених пројеката доводи до преоптерећења у бази. Зато клијент тражи да се пружи могућност брисања пројеката који су започели пре више од 5 година, а у међувремену су завршени. Ради планирања наредних корака и потребних људских ресурса, потребно је омогућити статистичко праћење података на годишњем нивоу.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

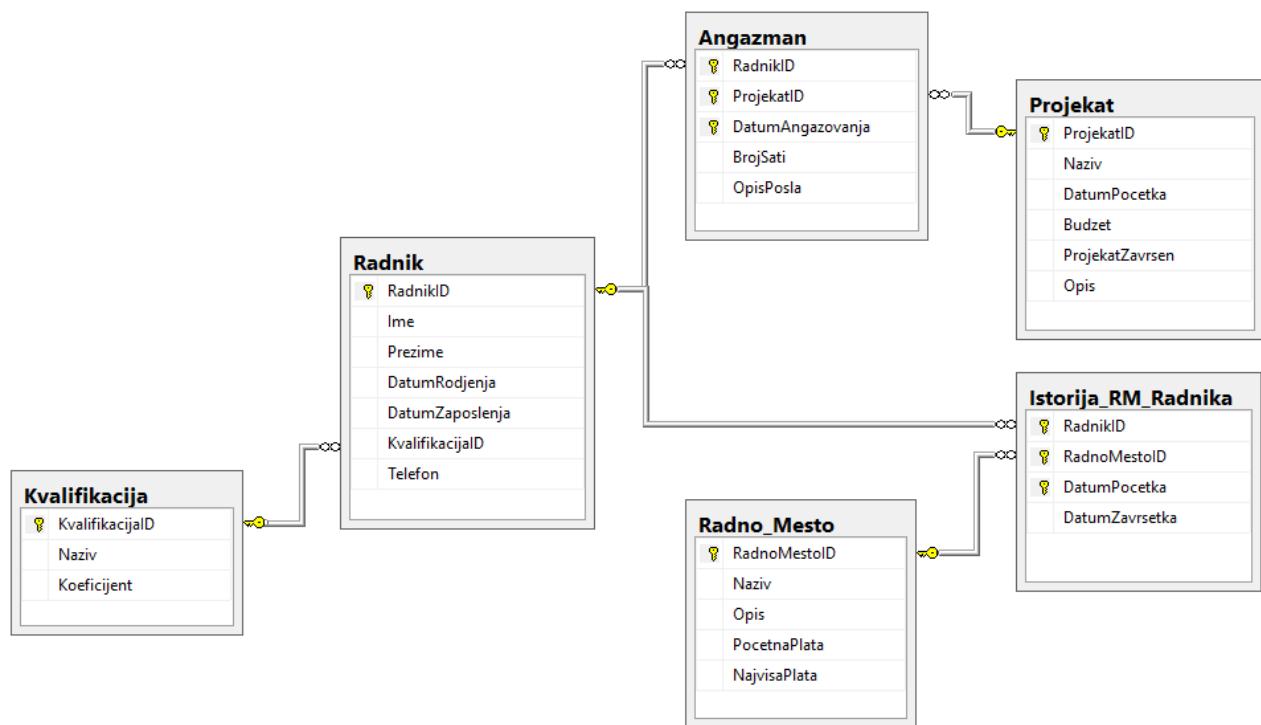
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A31**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A32**.

Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која податке о свим пројектима приказује у **listView** контроли.

Кликом на неки од редова, податке о пројекту треба приказати у осталим пољима форме.

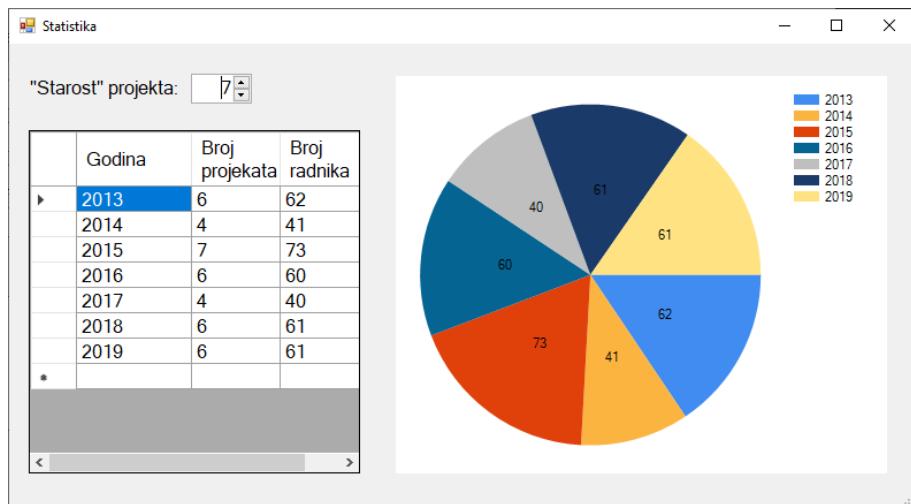
Кликом на дугме **Обриши**, обрисати пројекат чији су подаци приказани у пољима форме и после брисања променити приказ на форми (листа је освежи, а контроле у левом панелу форме добију подразумевани садржај – текст поља су празна, а **checkbox** неозначен).

Уколико се покуша брисање пројекта који не задовољава услов да буде обрисан, приказати одговарајућу поруку.

Креирати лог фајл који садржи списак шифара и назива обрисаних пројеката. Лог фајл назвати **log\_datum\_brisanja.txt** (нпр log\_31\_12\_2019.txt). Ако лог фајл већ постоји у радном директоријуму, само додати ново обрисане пројекте.

Sifra	Naziv	Datum pocetka	Budzet	Zavrsen	Opis
1	Izgradn...	20.09.2016	7465.55	True	doku.
2	Jačanje...	27.08.2012	27057....	False	doku.
3	Mehani...	17.05.2012	45997....	False	doku.
4	ICT psp	29.08.2012	23090....	False	doku.
5	EUNES	29.03.2014	42319....	True	doku.
6	Tempus	06.07.2018	33333....	True	doku.
7	Jačanje...	16.06.2019	24572....	False	doku.
8	Popis s...	15.09.2013	42342....	True	doku.
9	Amres	30.05.2019	4706.01	True	doku.
10	Progra...	20.06.2014	39244....	True	doku.
11	Progra...	12.06.2014	7601.98	False	doku.

Дугме **Преглед пројекта** отвара другу форму на којој се налази табела која у првој колони приказује календарску годину, а у наредне две број пројекта који су започели током те године и укупан број радника ангажованих на тим пројектима. Податке треба приказати само за пројекте који су „млађи“ од задатог броја година. Табела и график се приказују са сваком променом задатог броја година. Сортирати приказ по годинама почев од најмање.



Осмислити проширење базе којим се омогућава чување података о фирмама-подизвођачима ангажованим на пројекту. Због различите природе послова, поред сопствених запослених, клијент-носилац пројекта може да ангажује једног или више подизвођача. За свако ангажовање треба памтити од ког до ког датума је подизвођач ангажован на пројекту. При томе, директно у бази обезбедити да датум почетка буде мањи од датума завршетка пројекта. Подразумеване (*default*) вредности за датуме су текући датум за почетак, а за крај ангажовања последњи дан текуће године (нпр 31.12.2020)

Креирати дијаграм и бекап стања проширене базе.

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А4

### НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Сеоски туризам

Агенција која се бави развојем и организовањем сеоског туризма чува податаке у бази података чија је структура дата на слици у прилогу. Ширење туристичке понуде у селима свих региона у земљи доводи до потребе додавања нових региона, градова и села, као и допуну података оних који су већ унети. Планирање закупа капацитета подразумева анализу потражње, али и памћење неких додатних података у бази.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

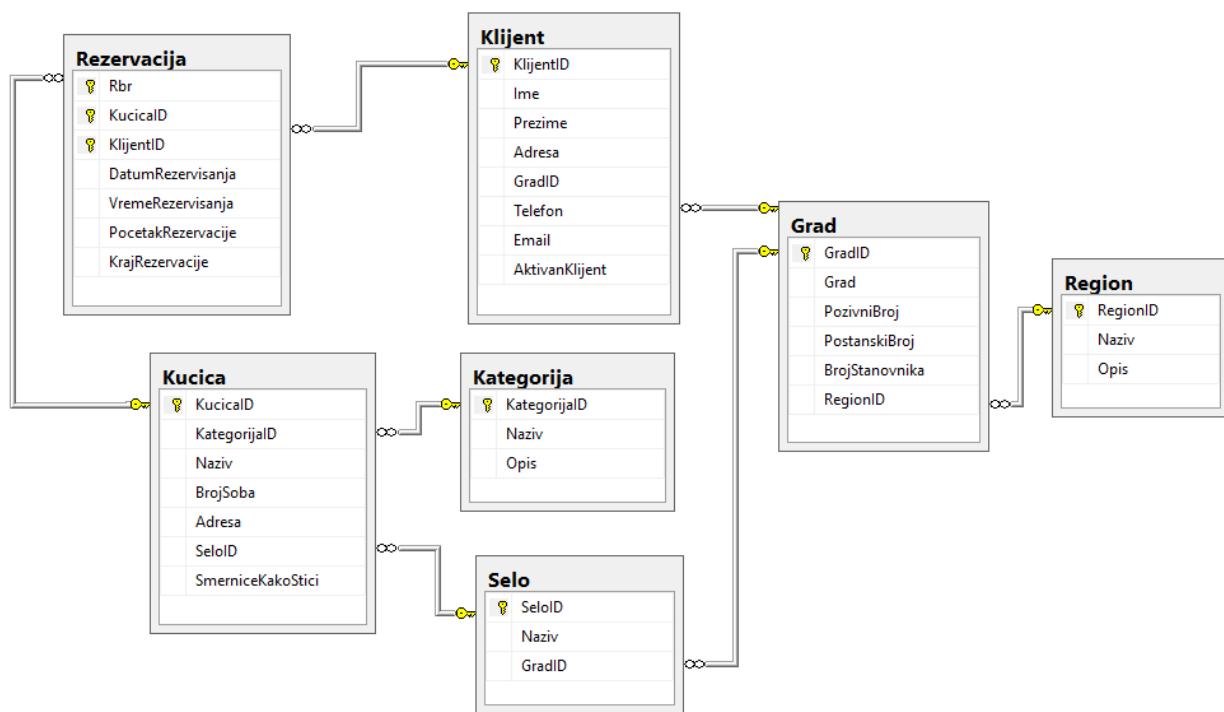
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A41**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A42**.

Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која податке о свим селима приказује у **listView** контроли. У **comboBox** треба учитати све градове сортирани по азбучном реду.

Уносом шифре села у одговарајуће текстуално поље, уколико за унету шифру постоје, подаци о селу се приказују на форми и селектује се одговарајући град у **comboBox-у**. Такође, одговарајући ред у листи бива селектован.

Подаци о изабраном селу се могу допунити и/или изменити уносом нових вредности, а кликом на дугме **Унеси измене**, измене вредности се памте у бази. У случају успешне измене освежити приказ у **listView** контроли.

Šifra	Selo	Grad
52	Badnjevac	Pirot
53	Žitište	Sombor
54	Bigrenica	Vlasotince
55	Bajmok	Novi Pazar
56	Bački Mon...	Niš
57	Kruščić	Vrbas
58	Laćarak	Pirot
59	Kumane	Niš
60	Dobrnje	Subotica
61	Bioska	Sombor
62	Novo Lani...	Novi Pazar
63	Ratkovac	Vršac

Unesite izmene >>>

Поруку о успешности измене приказати у **MessageBox**-у, а у случају неуспешне акције, додати запис у лог фајл са називом **error.txt**. Фајл садржи по један ред за сваку грешку насталу при изменама података (датум покушаја измене и разлог неуспешности промене).

Кликом на дугме **Излаз** у линији алата прве форме, затворити цељу апликацију.

Отворити кратко корисничко упутство кликом на дугме „**О апликацији**“.

Дугме **Додели ваучере** отвара другу форму на којој се за оне клијенте који су **активни** приказује пуно име клијента и број купљених аранжмана. Обезбедити да се на списку виде само они клијенти који имају **више од задатог броја** купљених аражмана.

Ime	Broj
Dimitrije Aleksić	19
Vanja Dačić	14
Đorđe Filipović	13
Milja Aleksić	13
Dunja Bajić	13
Balša Bogavac	12
Milan Bogavac	12
Petra Aleksić	12
Ana Dimitrijević	12
Anita Dimitrijević	12
Majda Dukić	12
Uroš Dukić	12
Dušanka Avramović	11
Kristina Borovac	11

Осмислити проширење базе којим се омогућава чување података о додатним понудама које су на располагању госту (интернет, ТВ, клима, базен, спа...). Једна кућница може нудити већи број додатних услуга и обрнуто, један тип услуге може бити на располагању у већем броју објеката. Потребно је памтити и у ком опсегу се креће цена услуге у конкретном објекту (кроз дизајн базе обезбедити исправан унос опсега цена, тј износ од< износ до)

Креирати дијаграм и бекап стања проширене базе.

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А5

### НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Продужени боравак

За вођење евидентије о пословању продуженог боравка креирана је база чија је структура дата на слици у прилогу. Како је понуда додатних активности које се организују за децу све богатија, потребно је креирати форму за преглед постојећих и додавање нове активности. Такође, због планирања термина појединих активности, потребно је направити анализу похађања појединих активности. Повећани обим послана намеће и потребу за проширењем базе података и чувањем података о којима до сада није вођена евидентија.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

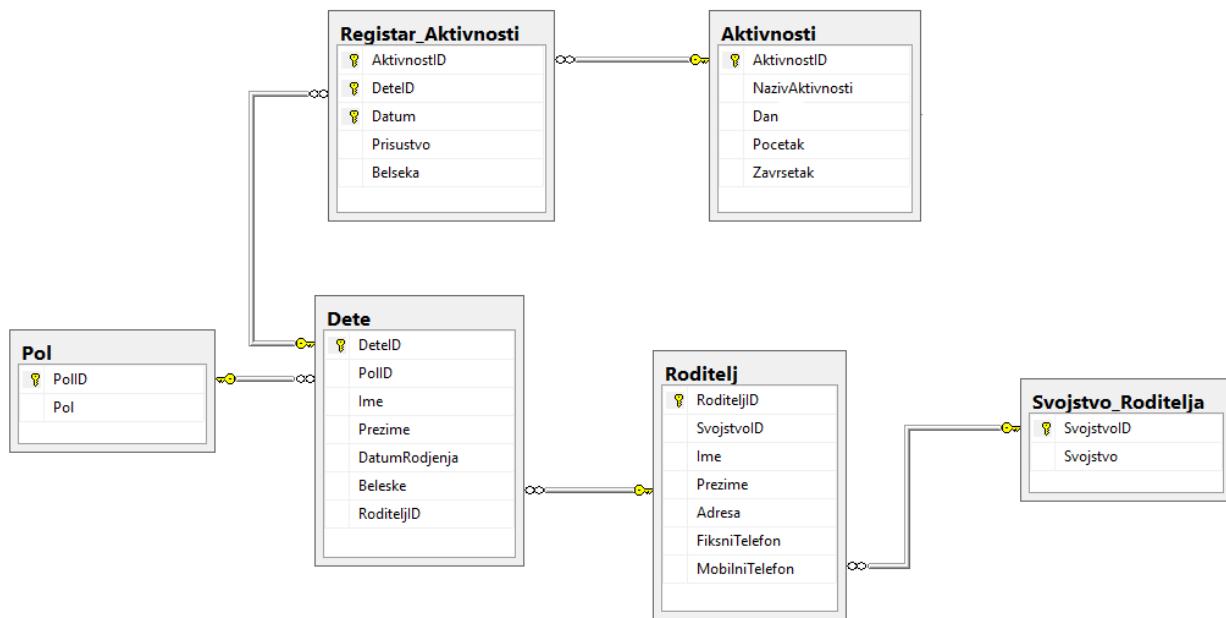
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A51**.

Резервна копија почетног стања базе дате је као **Прилог\_4ЕИТ\_A52**.

Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



На основу постигнутог договора наручиоца посла и развојног тима, потребно је урадити следеће:

Креирати форму на којој се у сваком моменту виде све активности. Подаци се приказују у **listView** контроли.

Кликом на неку од ставки **listView** контроле, податке о селектованој активности треба приказати у осталим контролама форме.

Кликом на дугме **Унеси** уписује се нова активност и при томе освежава приказ у **listView** контроли. За сваку планирану активност потребно је обавезно при самом креирању унети назив активности, а термин одржавања (дан у недељи и време) може бити и касније одређен.

Клик на дугме **Изађи** на првој форми, затвара целу апликацију.

Линк „**О апликацији**“ (на менију прве форме) треба да отвори кратко корисничко упутство.

The screenshot shows the 'Aktivnost' form with the following details:

Šifra	Naziv	Dan	Pocetak	Završetak
15	Crtanje	Ponedeljak	12:00	13:00
21	Setnja	Ponedeljak	14:00	18:00
36	Recital	Ponedeljak	09:00	15:00
38	folklor	Utorak	10:00	12:00
22	Koncert	Utorak	11:00	15:00
16	Pozoriste	Utorak	07:00	11:30
17	Sport	Sreda	12:00	14:00
13	Engleski	Sreda	09:00	11:00
37	Gluma	Sreda	15:00	17:00
18	Plivanje	Cetvrtak	07:00	15:00
19	Hor	Petak	08:00	11:30
14	Francuski	Petak	13:00	15:00

On the left side of the form, there are input fields and dropdown menus:

- Šifra: 17
- Naziv: Sport
- Dan u nedelji: Sreda
- Vreme početka: 12:00
- Vreme završetka: 14:00
- Buttons: Unesi (highlighted in blue), Izadi

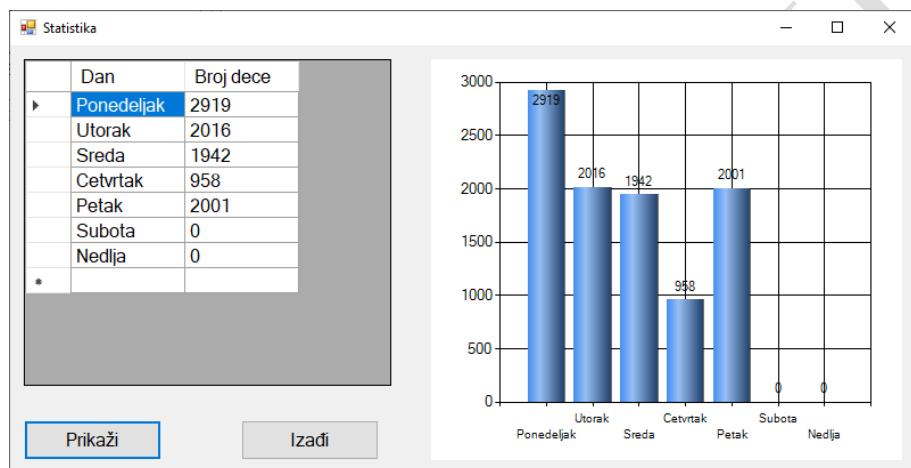
Креирати другу форму која се отвара кликом на дугме са иконом графика у линији са алатима на првој форми.

На другој форми, кликом на **Прикажи**, табеларно и графички приказати колико деце током сваког од дана у недељи присуствује организованим активностима.

Дугме **Изађи** на другој форми, затвара само другу форму и враћа се на почетну.

Осмислите проширење и креирати додатне табеле, везе и ограничења у бази који ће омогућити да се у бази чувају и подаци који се тичу наставника-васпитача одговорних за координацију организацијом активности. За сваку активност треба памтити ко од васпитача је координатор за ту активност. Такође је потребно памтити и датум пријема детета у продужени боравак (обезбедити да датум пријема не може да буде старији од датума рођења)

Креирати дијаграм и бекап стања проширене базе.



## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А6

### НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Половни аутомобили

Агенција за продају половних аутомобила исказала је потребу за автоматизацијом претраге расположивих возила. На конкурсу је ваша фирма изабрана за извођача посла и поделом задужења добили сте задатак да реализујете део апликације који се односи на измену података о аутомобилима у бази (у случају када се због иницијалног погрешног уноса јавила потреба за корекцијом), као и модул за проверу броја расположивих возила у оквиру задатих критеријума пређене километраже. Поред продаје откупљених возила, агенција почиње да се бави и посредовањем у продаји, тако да је потребно проширити саму базу могућностима чувања неких додатних података.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

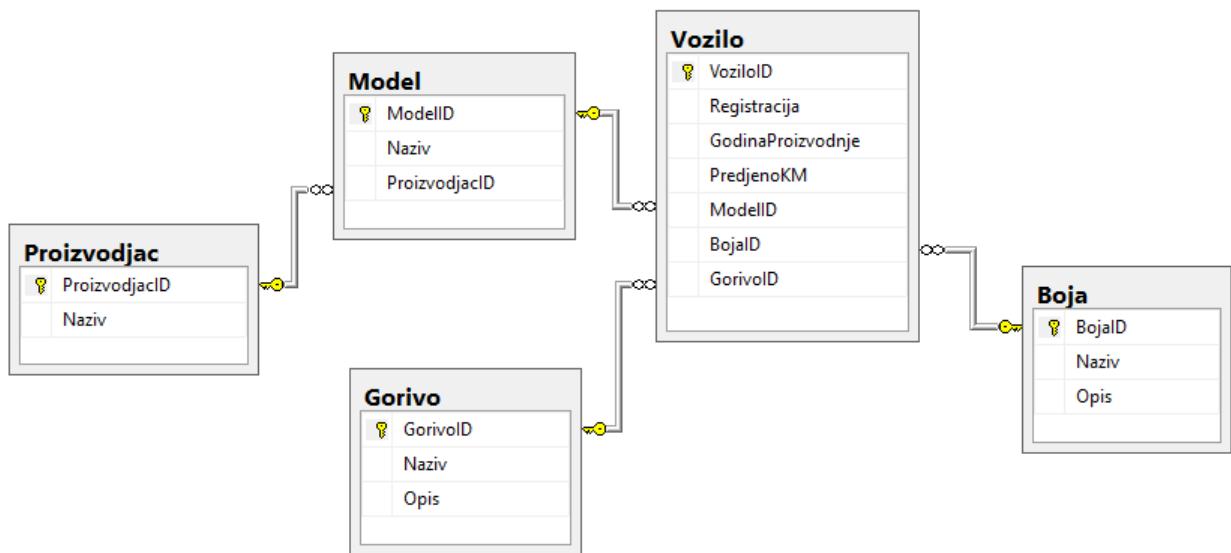
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A61**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A62**.

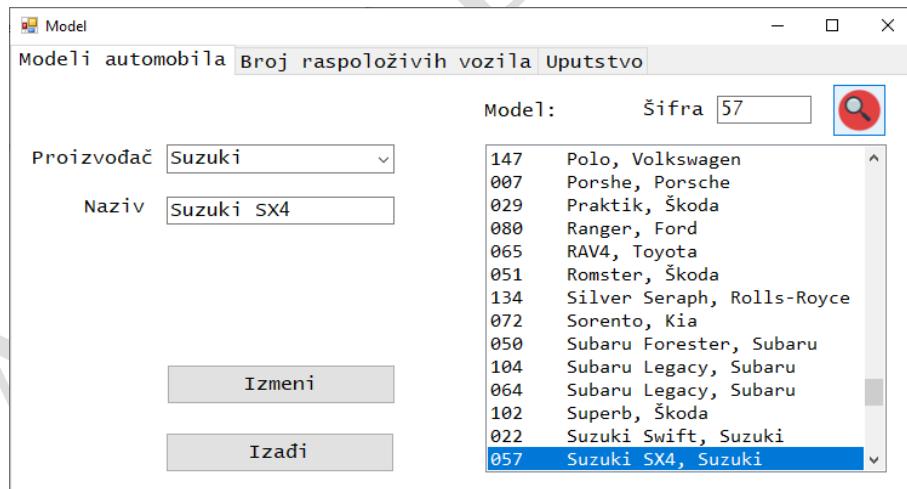
Архива слика и икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



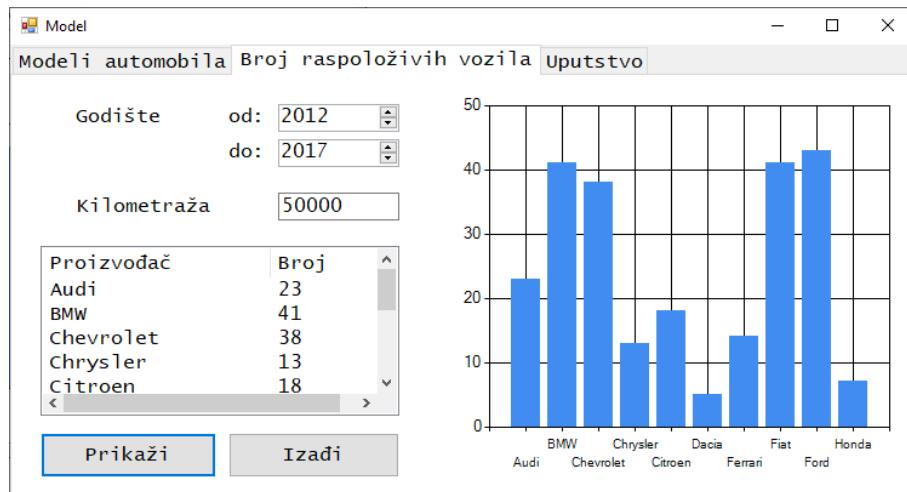
Креирати форму која има три таба. На првом табу се приказују подаци о свим расположивим моделима аутомобила. Подаци о моделима аутомобила приказује се у *listBox* контроли.

На почетку ни један модел није селектован, а уносом шифре у одговарајуће поље и кликом на дугме са иконом лупе у листи се проналази модел са унетом шифром, селектује се и подаци о њему приказују у левом делу форме. Ако модел са унетом шифром не постоји, селекција се склања и контроле у левом делу форме се ресетују на иницијалне вредности.



Уколико су производач или назив селектованог модела погрешно унети, могу се изменити уносом нових вредности. Кликом на дугме **Измени**, врши се корекција вредности у бази. Ако је измена успешно одрађена, освежити листу и селектовати изменјен модел. Ако измена није начињена, послати поруку кориснику.

Дугме **Изађи** затвара целу апликацију.



Други таб форме организовати као на скици. Кликом на дугме **Прикажи**, потребно је приказати расположиви број возила сваког од производјача. Треба разматрати податке само за возила чије је годиште производње у задатом интервалу година и која су максимално прешла дату километражу.

На трећем табу је исписано кратко корисничко упутство и документација о апликацији.

Због бављења посредовањем у продаји, у базу је потребно додати могућност да се чувају подаци о тренутним власницима аутомобила (име, презиме, телефон, адреса, да ли се ради о првом власнику). Сваки аутомобил може да има само једног власника. Такође је потребно додати могућност памћења почетне понуђене цене за возило, са тим што треба обезбедити да се не може унети нелогична вредност (цена не може бити мања од нуле).

Креирати дијаграм и бекап стања проширене базе.



ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А7

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Факултетска евиденција**

Факултет води евиденцију о томе које испите студенти слушају, да ли су одслушали и одрадили предиспитне обавезе, као и о консултацијама које су обавили са професорима. .

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

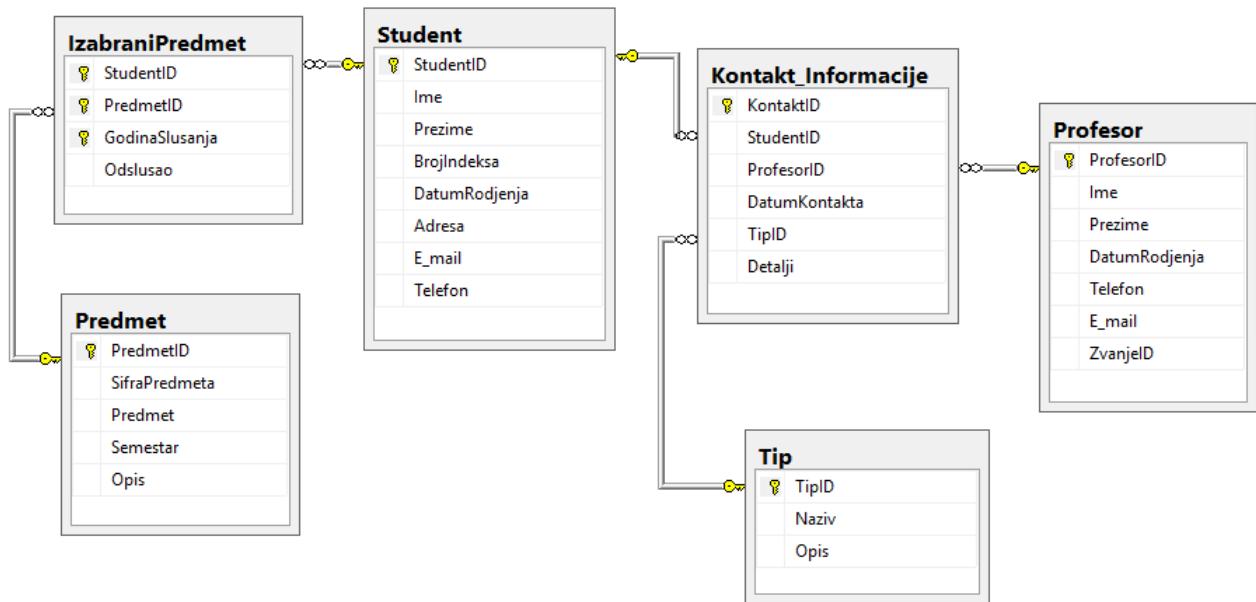
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A71**.

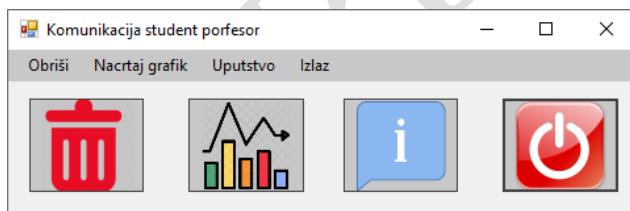
Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A72**.

Архива слика и икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Почетна форма само служи да се са ње бира која од преостале две форме треба да буде позвана.



Кликом на кантицу отвара се форма за брисање предмета из базе. При учитавању ове форме сви предмети се приказују у контроли **dataGridView**, а кликом на неку од линија у овој контроли врши се селктовање предмета кога треба обрисати, при чему се подаци о том предмету приказују у левом панелу форме.

The screenshot shows a window titled 'Predmet' with a left panel for entering subject data and a right panel for displaying a list of subjects.

**Left Panel (Predmet):**

Predmet ID	4
Šifra (kratko ime)	9A7GR
Predmet	Matematika III
Semestar	3
Opis	Plan i program predmeta u trećem semestru.....

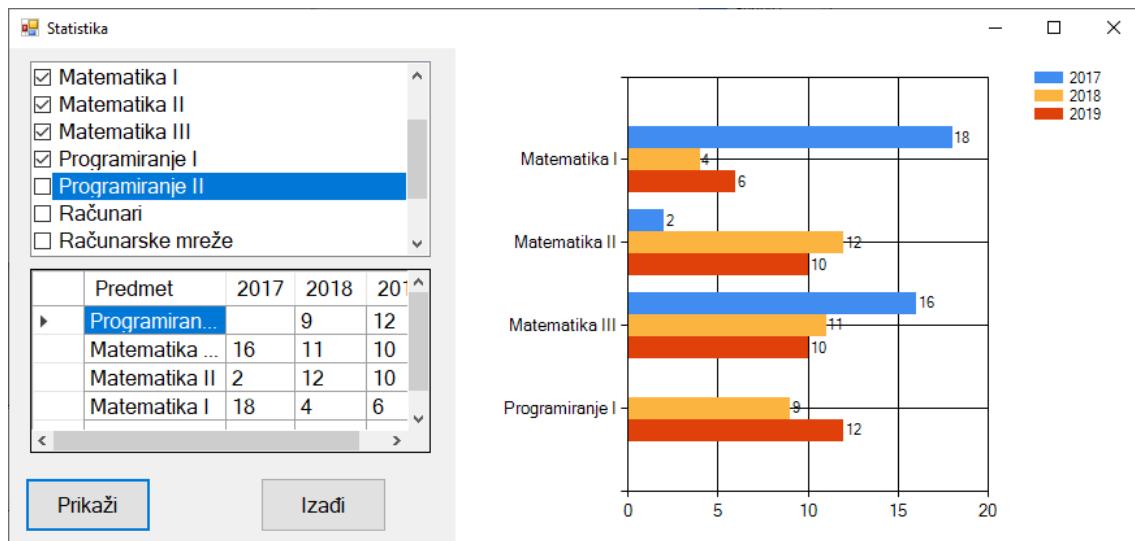
**Right Panel (Predmeti):**

ID	Sifra	Naziv	Semest	O
1	CS5WT	Fizika	2	
2	5FIR8	Programiranje I	6	W
3	XDVAS	Filozofija	4	A.
4	9A7GR	Matematika III	3	W
5	MMMM	Matematika II		
7	74LL2	Programiranje II	8	V.
8	YI21N	Računari	4	4..
11	TWCG4	Računarske m...	5	O.
12	AHVML	Web dizajn	7	T..
13	BBBBB	Baze podataka	6	T..
14	PT1WM	Web programir...	5	H..
15	3E2MW	Matematika I	6	F

Buttons at the bottom: 'Obriši' (Delete) and 'Izađi' (Exit).

Дугме **Обриши**, брише селектовани предмет и обавештава о успешности брисања. После брисања треба ажурирати списак у **dataGridView**-у. Дугме **Изађи** затвара форму и враћа се на почетну.

Друга форма се отвара са почетне кликом на икону графика. У **checkListBox**-у су излистани сви предмети и служи да се означе предмети за које је потребно приказати податке (упоредно се посматра не више од 5 предмета). Дугме **Прикажи** приказује број студената који су током претходне 3 календарске године слушали изабране предмете. Податке приказати на дијаграму. Дугме **Изађи** затвара форму.



Треће (инфо) дугме почетне форме отвара кратко корисничко упутство, а четврто дугме затвара целу апликацију.

Потребно је базу проширити тако да се води евиденција и о студијским групама (смеровима). Памтити шифру, назив и датум акредитације програма. Програм мора да буде акредитован и датум акредитације не сме да буде старији од 7 година.

Један предмет се може полагати на већем броју студијских програма. Такође, потребно је памтити податке и о томе на којој студијској групи студент студира.

Креирати дијаграм и бекап стања проширене базе.

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А8

### НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Археолошка налазишта

Друштво археолога води евидентију о налазиштима на којима су радили током година, као и о проналасцима који су се десили. Да би се извршила класификација и систематизација проналаска они се разврставају у различите типове тј. категорије и поред тога се памти и период из кога се сматра да проналазак потиче.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

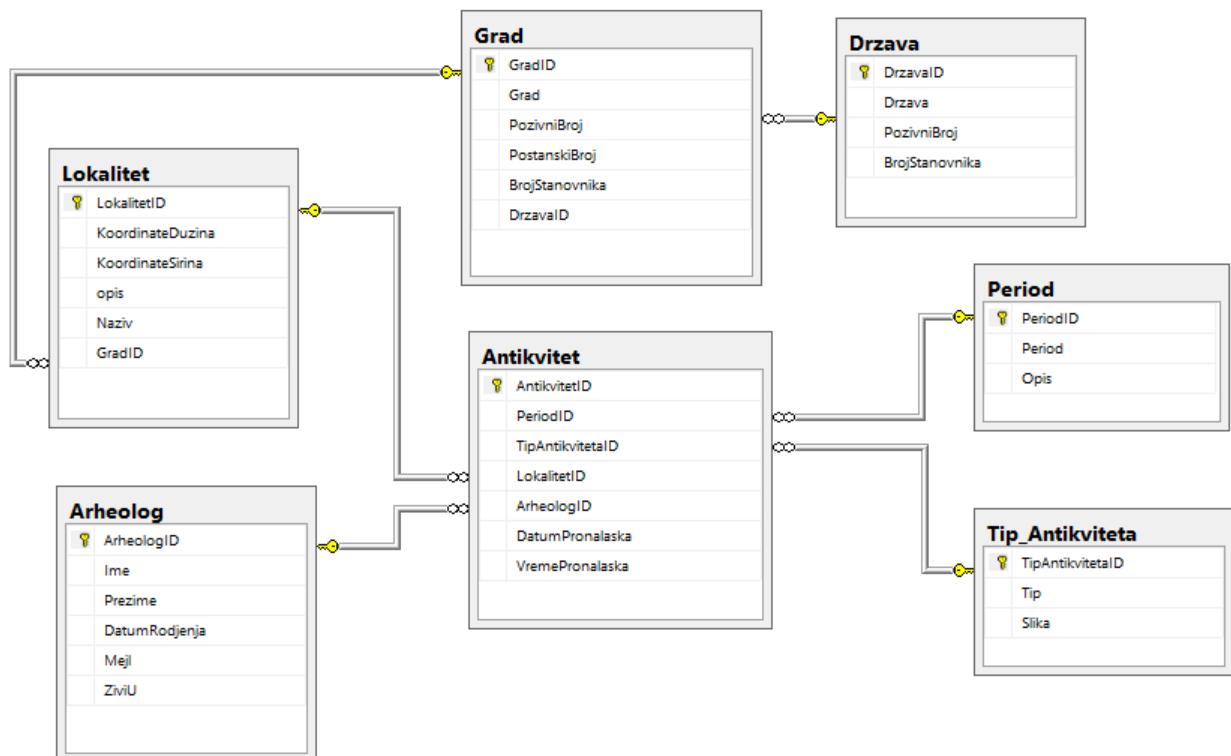
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A81**.

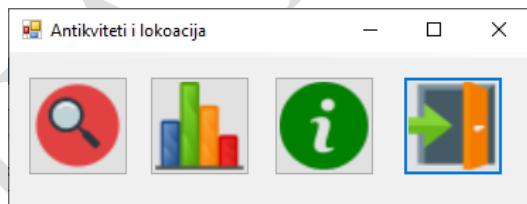
Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A82**.

Архива слика и икона дата је као **Прилог\_4ЕИТ\_A00**.

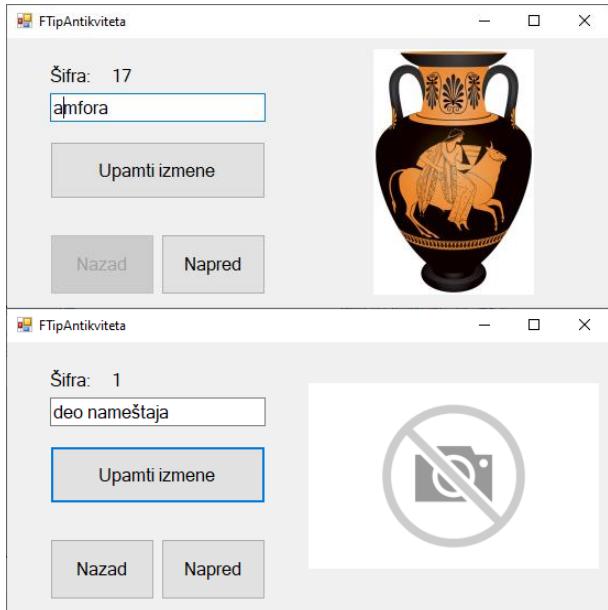
## ПРИЛОГ:



Почетна форма само служи да се са ње бира која од преостале две форме треба да буде позвана.

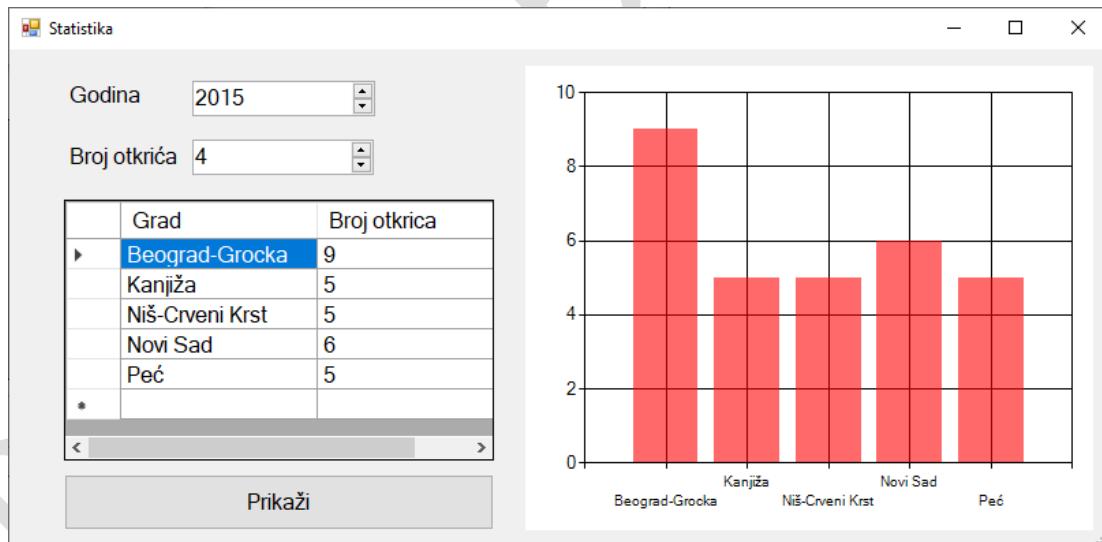


Кликом на лупу отвара се форма за преглед различитих категорија (типова) антиквитета који се памте у бази. За унети тип се памти назив и релативна путања до одговарајуће слике, а шифру под којом је тип запамћен аутоматски креира сам база при уносу. Слике су запамћене у подфолдеру радног фолдера апликације у облику "**antikviteti/imeFajla**". Уколико није доступна слика, на њеном месту се налази путања до подразумеваног фајла "**antikviteti/no\_image.jpg**".



Кликом на дугмад са ознакама **Напред** и **Назад** врши се листање доступних типова. Клик на слику отвара **openFileDialog** за избор нове слике. Изабрана слика је прикаже у **pictureBox**-у. Памћење измене (слике или назива типа антиквитета) у базу се врши кликом на **Упиши измене**.

Друга форма се отвара са почетне избором иконе са приказом графика. Кликом на дугме **Прикажи** друге форме треба приказати називе свих градова у чијој је околини било више од **датог броја открића**, после неке **задате године**, као и одговарајући број открића. Податке приказати на форми у **dataGridView** контроли и на графику.



Треће (инфо) дугме почетне форме отвара кратко корисничко упутство, а четврто дугме затвара целу апликацију.

Да би се водила евидентија о томе где се налазе пронађене вредности, потребно је памтити у ком музеју (назив, сајт, мејл адреса, телефон) се тренутно налази свака од пронађених вредности. Такође је потребно и регистровати у ком се граду музеј налази.

Креирати дијаграм и бекап стања проширене базе.

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А9

### НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Фудбалски клубови

Међународни фудбалски савез води евиденцију о одржаним такмичењима, клубовима-учесницима такмичења, као и о постигнутим резултатима на одиграним утакмицама. За сваки клуб се у бази памти из које државе и града клуб долази, као и основни подаци о његовом стадиону.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

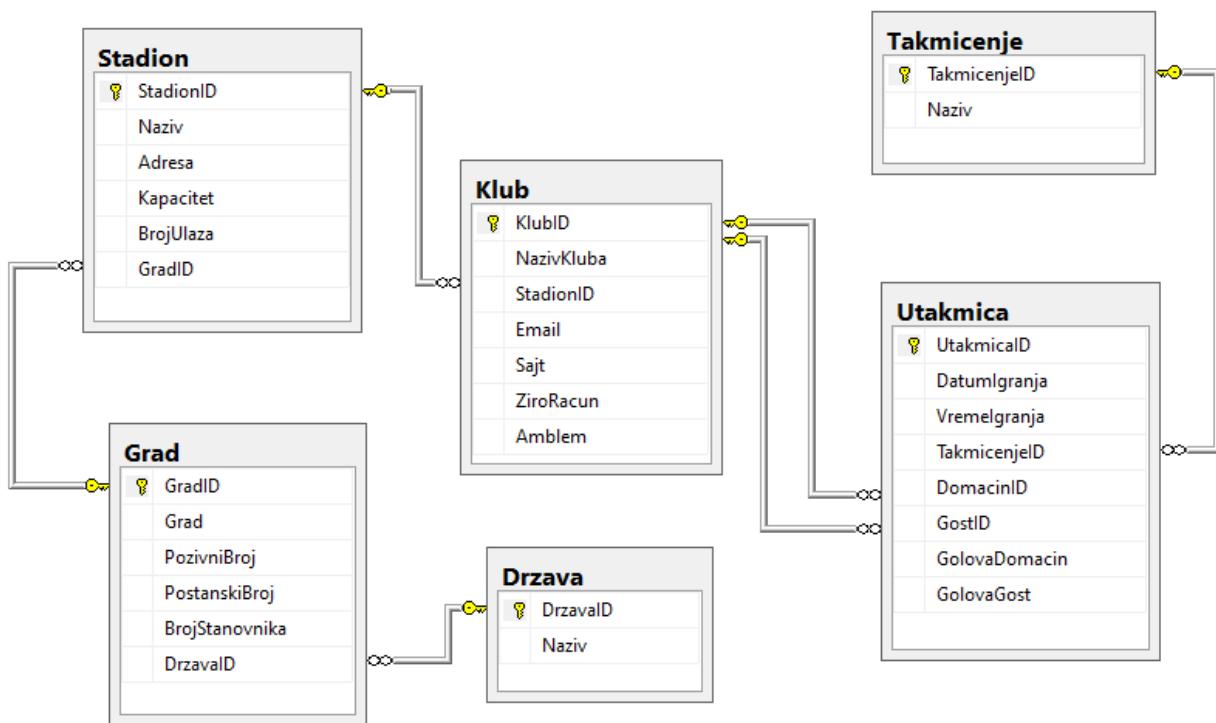
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A91**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A92**.

Архива слика и икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:

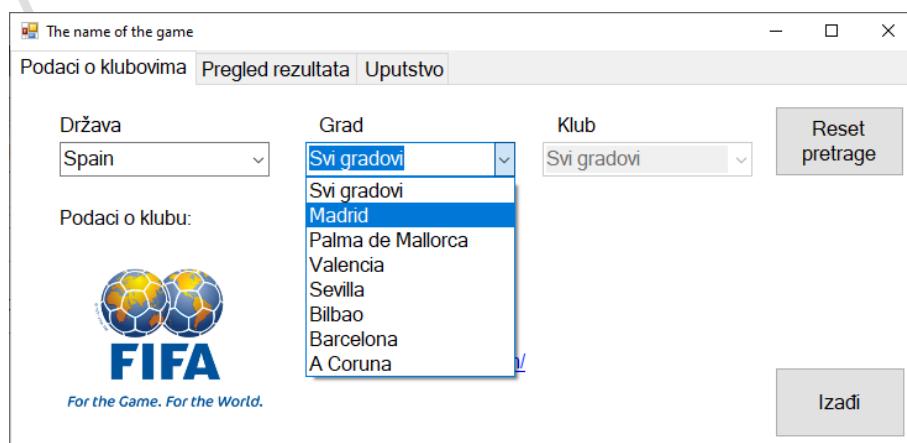


На првом табу форме врши се преглед података о фудбалским клубовима. Када се из првог **comboBox-a** изабере држава, онда се отвара други у коме се приказују само градови изабране државе. Када се изабере град, отвара се трећи comboBox где се излиставају само клубови из изабраног града. Коначно, избором клуба, у контролама испод линије за претрагу се приказује амблем клуба и подаци о томе из ког је града и на ком домаћем стадиону игра у свом граду.

Када се избор у comboBox-у градова врати на прву ставку **Сви градови**, поље клубова се опет затвара. Аналогно, када се у пољу држава изабере **Све државе**, падајућа листа градова се затвара.

Амблем клуба се памти као релативна путања до слике. Слике су запамћене у подфолдеру радног фолдера апликације у облику **klubovi/imeFajla**. Уколико није доступна слика, на њеном месту се налази путања до подразумеваног фајла **klubovi/no\_image.jpg**.

Кликом на **Ресет претраге**, стање падајућих листа за избор се враћа на почетно, а дугме **Изађи** затвара апликацију.





Други таб служи за дневни преглед утакмица. Потребно је изабрати датум за који се тражи преглед стања (**dateTimePicker**), а затим кликом на дугме **Прикажи**, у **listView** треба приказати све утакмице одигране тог дана и то: назив стадиона на коме је утакмица одиграна, име домаће екипе, резултат домаћина, резултат госта и име гостујуће екипе. Дугме **Изађи** затвара апликацију.

Podaci o klubovima		Pregled rezultata	Uputstvo	Prikaži	Izađi
Datum utakmice	01.05.2019				
Stadion	Domaća ekipa	Golovi do...	Golovi go...	Gostujuća ekipa	
Estadi Ciutat de Valencia	Levante	0	2	Sevilla	
Stadio Ennio Tardini	Parma	1	1	Lazio	
San Siro	Milan	2	1	Juventus	
Camp Nou	Barcelona	4	2	Levante	
Estadio de Riazor	Deportivo de La Coruna	3	0	Bologna	
Estadio Santiago Berna...	Real Madrid	0	0	Roma	

Трећи таб (**Упутство**) садржи кратко корисничко упутство и објашњење структуре апликације.

Потребно је базу проширити могућношћу памћења играча. За играча треба памити име, презиме, датум рођења и из које земље је пореклом. Играч у различитим периодима може да игра за различите клубове, па се зато памти датум када је почeo и датум до када је играо за неки тим. Први датум је обавезан, а други је празан уколико играч још увек игра за тај тим. У самој бази треба обезбедити да други датум или није постављен или је мањи од првог датума.

Креирати дијаграм и бекап стања проширене базе.

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А10**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Риболовачко друштво**

Риболовачко друштво пеџароша „Рибар“ располаже базом података у којој се води евиденција о свим пеџарошима и местима за пеџање у Републици Србији. Друштво „Рибар“ жели да освежи податке који су уписаны у њиховој бази. Такође су одлучили да помогну својим члановима у избору риболовачке опреме, тако што ће проширити своју базу са вођењем евиденције о опреми коју су користили за сваки улов. Апликацији је потребно додати функционалности за преглед и корекцију података о пеџарошима као и графички приказ извештаја о улову пеџароша у одеђеном временском периоду.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_А10**, где се као име и презиме уноси име кандидата.

Предвиђено време за израду задатка је 180 минута.

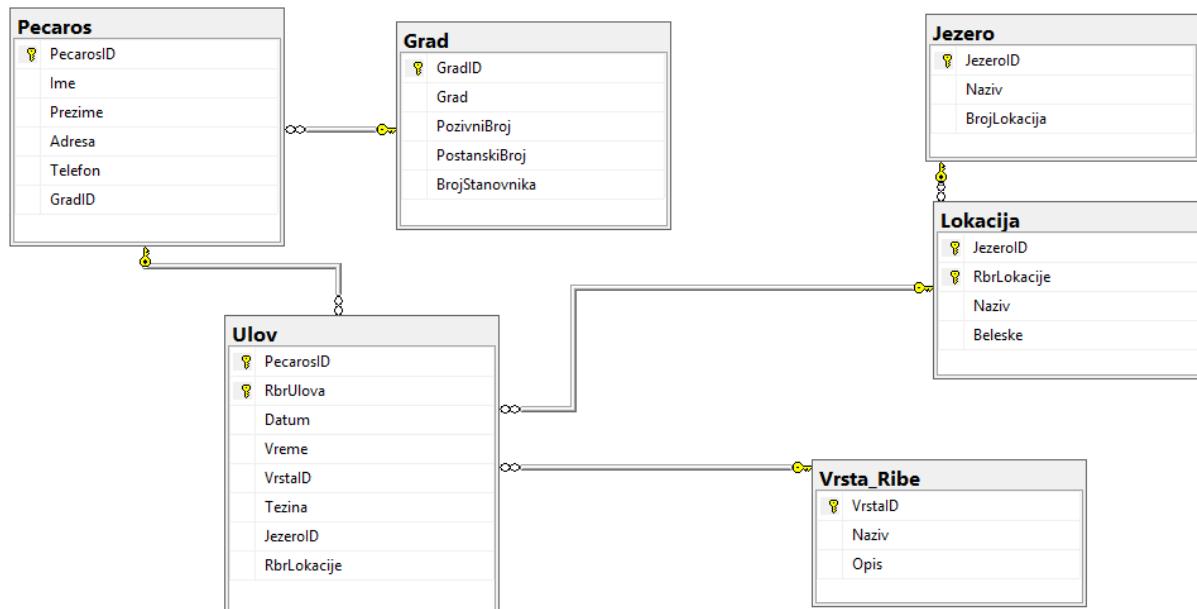
По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка кандидат може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_А10\Потребни ресурси**.

## Прилог за задатак:

Риболовачко друштво је приложило дијаграм за тренутну базу:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати почетну форму са које се кликом на ставке менија приступа појединачним функционалностима апликације.

Креирана форма поред менија треба да приказује и податке о свим пеџарошима у **listBox** контроли (шифра, име презиме, адреса, град и телефон). Такође, на форму треба поставити поља за појединачни приказ изабраног пеџароша. Појединачна поља иницијално треба попунити са подацима пеџароша, који има најмању вредност шифре. Одабир пеџароша, чије податке је потребно променити, врши се селекцијом реда у **listBox** контроли. Селектоване податке приказати у појединачним пољима форме. Све вредности које су приказане у пољима за појединачни приказ пеџароша, сем шифре, су подложна промени. Омогућити да у **comboBox** контроли, која омогућава избор града, буду учитани сви градови. Називе градова сортирати по абецедном поретку. Кликом да дугме менија **Izmeni** све вредности поља о приказаном пеџарошу се уписују у базу. Ажурирањем податка у бази освежити и приказ у **listBox** контроли. Поруку о успешности измене података приказати у **MessageBox**-у.

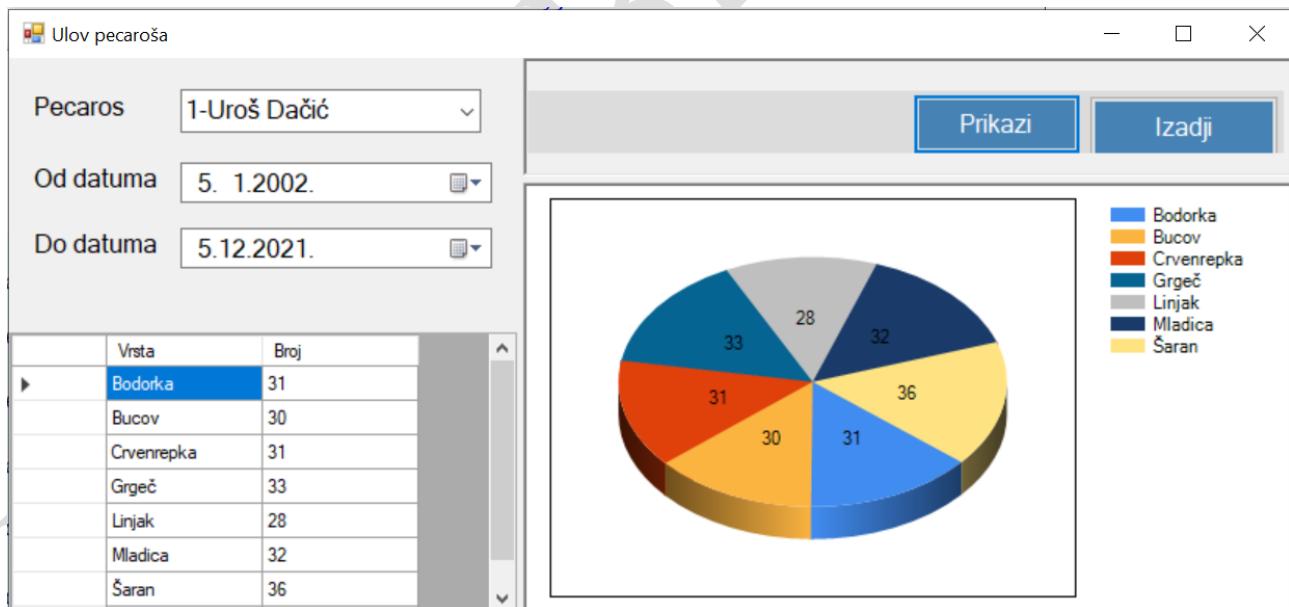
Функционалност менија:

- Ставка менија **Izmeni** омогућава ажурирање података о пеџарошу;
- Ставка менија **Analiza** приказује форму са графиконом.
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство.
- Ставка менија **Izlaz** затвара апликацију.

Дизајн креирање форме:

Кликом на ставку менија **Анализа**, отвара се форма за графички приказ укупног броја риба, које је уловио пециар у одређеном временском интервалу. Форма треба да садржи поље за одабир пециара, шифру и име пециара. За приказ и одабир пециара користити контролу **ComboBox**. У контроли **ComboBox** треба приказати шифру, име и презиме пециара. Приказани подаци су сортирани по шифри у растућем редоследу. Иницијална вредност у контроли **ComboBox** је „Izaberi pecaroša“.

Оба датума морају бити дефинисана. Кликом на дугме **Prikazi** резултати упита се приказују у контроли за табеларни приказ података и на основу тих података се црта график. Дугме **Izadji** затвара отворен образац и враћа на почетну форму.



Осмислите проширење базе којим се омогућава чување података о пециаршкој опреми. Потребно је креирати табелу опрема (штап, машиница, најлон, мамац, удица) и у табелу улов додати опрему коју је пециар користио у том улову.

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А11**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Евиденција књига**

Музеј града Новог Сада има приватну колекцију књига за потребе својих запослених. Ради лакшег вођења евиденције и чувања података креирали су базу података чија је структура дата на слици у прилогу. За постојећу базу потребно је направити апликацију за преглед и упис нових аутора као и графички приказ броја књига по аутору. У циљу евиденције коришћења књига од стране запослених потребно је проширити постојећу базу са новим подацима.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_11**, где се као име и презиме уноси име кандидата.

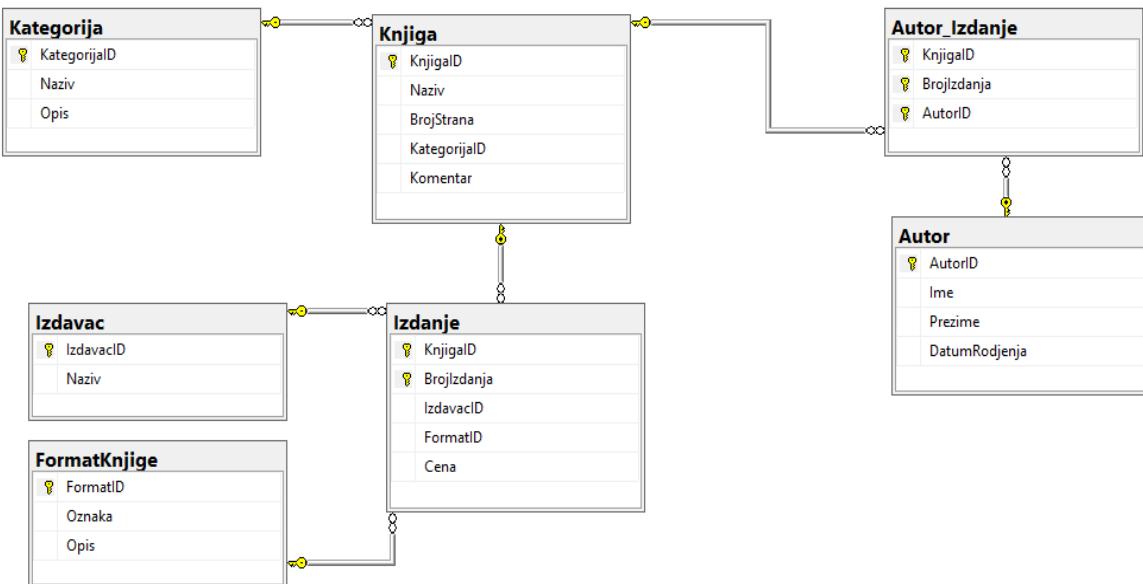
Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка кандидат може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A11\Потребни ресурси**.

#### **Прилог за задатак:**



Развојним планом апликације предвиђено је да се уради следеће:

Креирати почетну форму са које се кликом на ставке менија приступа појединачним функционалностима апликације.

Креирана форма поред менија треба да приказује и податке о свим ауторима у **listBox** контроли (шифра, име презиме, датум рођења). Такође на форму треба поставити поља за појединачни приказ изабраног аутора и дугмад **RESET**, **UPIŠI** и **ODUSTANI**. Активно је само дугме **RESET**. Појединачна поља иницијално треба попунити са подацима аутора који има најмању вредност шифре. Након селекције реда у **listBox** контроли или уноса шифре у поље за шифру, подаци о аутору се приказују у одговарајућим пољима форме. Ако непостоји аутор са унетом шифром, приказати одговарајућу поруку. У случају да се обрише садржај поља за шифру, обрисати и садржај свих појединачних поља. Приликом уноса новог аутора поље за шифру није потребно попунити, јер се приликом уписа аутора у базу аутоматски генерише вредност шифре.

За приказ датума користити контролу која је приказана на приложеној слици форме **Unos autora**. Кликом на дугме **RESET** брише се садржај свих поља и припрема се форма за упис: поље шифра се не може мењати и активна су дугмад **UPIŠI** и **UDUSTANI**. Кликом на дугме **UPIŠI** врши се упис аутора у базу при чему се освежава приказ у **listBox** контроли. Након уписа дугмад **UPIŠI** и **ODUSTANI** постају поново неактивна. Такође, кликом на дугме **ODUSTANI** се прекида упис и дугмад **UPIŠI** и **ODUSTANI** постају поново неактивна. Поруку о успешности уписа приказати у **MessageBox**-у.

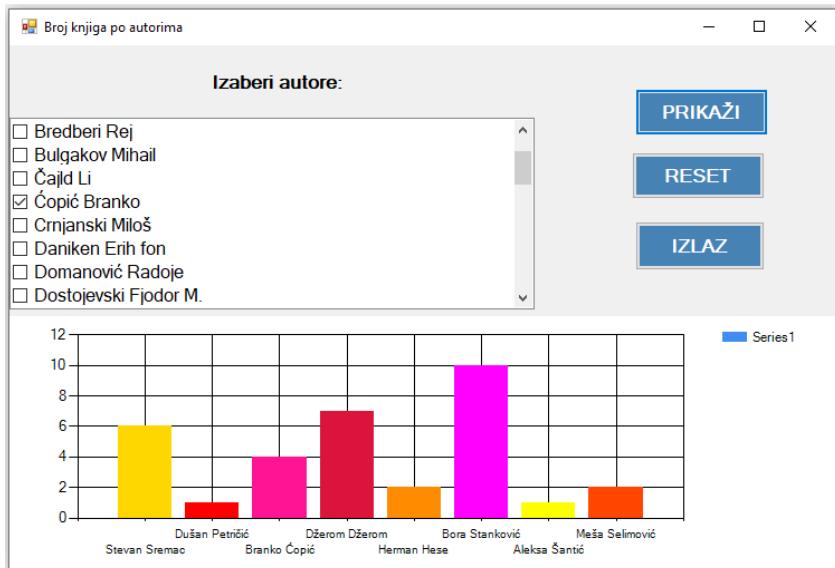
Функционалност менија:

- Ставка менија **Analiza** приказује форму са графиконом.
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство.
- Ставка менија **Izlaz** затвара апликацију.

ŠIFRA	IME	PREZIME	DATUM
29	Mihail	Bulgakov	01.01.1921
30	Mirjana	Durđević	29.04.1947
31	Meša	Selimović	01.01.0001
32	Li	Čajld	29.11.1975
33	Radoje	Domanović	05.09.1922
34	Lav Nikolajević	Tolstoj	30.12.1909
41	Sasa	Saskovic	06.12.2021
42	Sasa	Saskovic	06.12.2021
43	Milan	Bozic	01.01.2012
44	Milan	Pena	27.09.1938
45	Bojan	Bojanovic	26.07.2021

Ставка менија **Анализа**, графички приказује укупан број књига које је издао тражени аутор. За приказ и одабир аутора користити checkedListBox. Имена и презимена аутора треба да буду сортирана по азбучном реду. Корисник бира најмање два аутора.

Дугме **Изађи** затвара отворен образац, и враћа на почетну форму.



Осмислiti проширење базе којим се омогућава праћење података о запосленима (име,презиме, функција) и употреби(кад је узета књига, кад је књига враћена). Вредност датума повратка мора бити после датума кад је књига узета. Потребно је водити рачуна да једну исту књигу запослени може узимати више пута. Такође је потребно табелу **Izdavac** проширити са пољем адреса.

РАДНА ВЕРСИЈА

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А12

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Фудбалски стадиони**

Међународни фудбалски савез води евиденцију о одржаним такмичењима, клубовима-учесницима такмичења, као и о постигнутим резултатима на одиграним утакмицама. За сваки клуб се у бази памти из које државе и града клуб долази, као и основни подаци о његовом стадиону.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

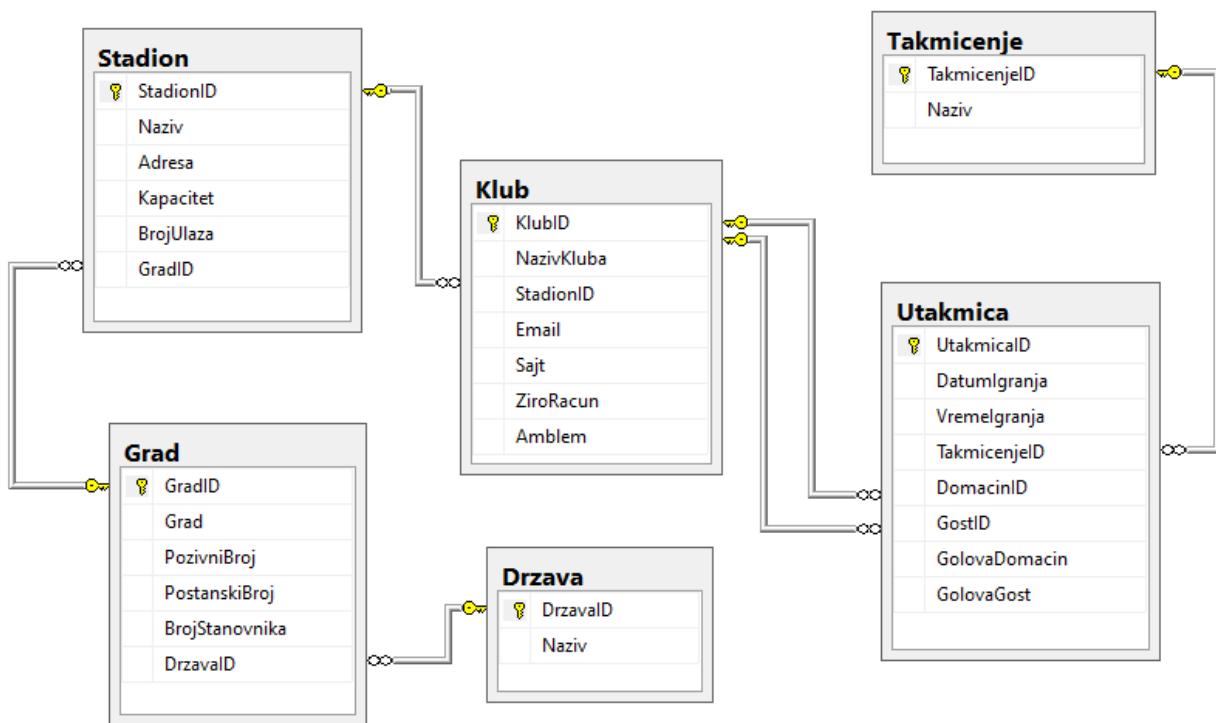
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A121**.

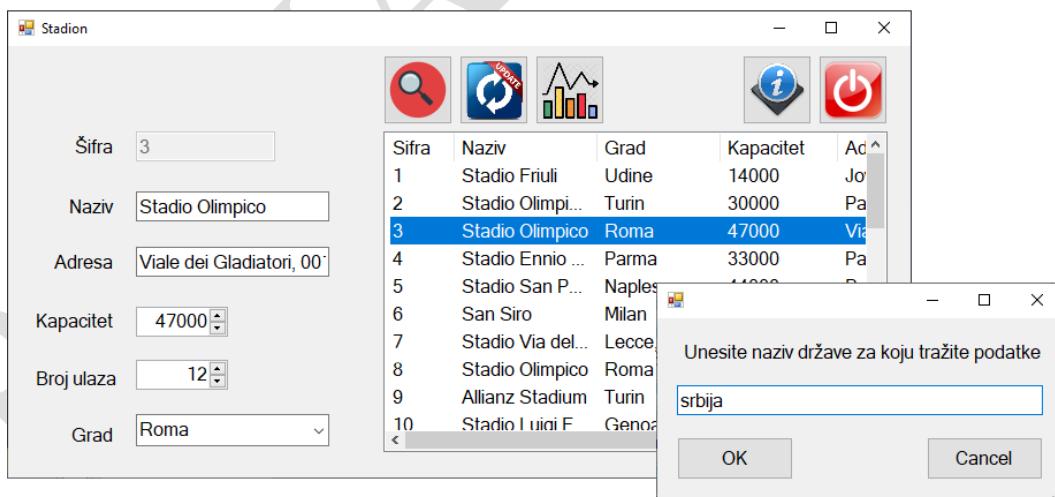
Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A122**.

Архива слика и икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



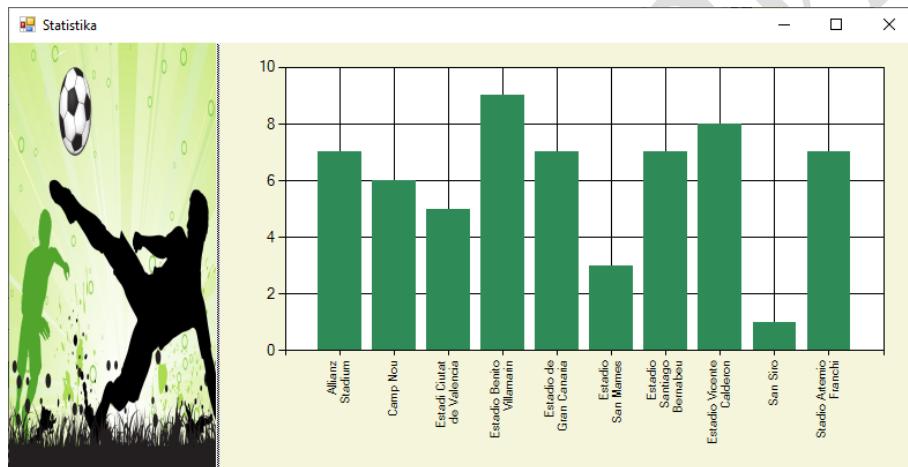
На почетној форми врши се преглед података о фудбалским стадионима. По отварању сви стадиони су видљиви у **listView** контроли. Када се кликне на неки од редова у табеларном приказу, подаци о кликнутом стадиону се показују у левом панелу форме. У случају да су погрешно унети, подаци се могу кориговати и ажурирати у бази кликом на дугме са **Update** иконом. По успешној измени приказ у **listView** контроли бива освежен.



Ради лакшег проналажења траженог стадиона, могуће је сужење претраге кликом на дугме са иконом лупе. Отвара се форма са једним текст пољем где се задаје назив државе чије стадионе треба приказати. Кликом на **OK** примењује се филтер или ако је поље остало празно, враћа се на приказ свих стадиона. Кликом на **Cancel** одустаје се од филтрирања приказа. Дугме са ознаком **Инфо**, отвара кратко упутство и приказ структуре апликације. Дугме **Изађи** затвара апликацију.

Sifra	Naziv	Grad	Kapacitet	Adre
98	Čair	Niš	19000	9. bri
99	Stadion Partizan	Beograd	32710	Hum
100	Stadion Crvena Zvezda	Beograd	55538	Beo
101	Omladinski stadion	Beograd	20000	Karađ
102	Stadion Karađorđe	Novi Sad	15000	Novo
103	Stadion FK Šabac	Kraljevo	5000	Kralje
104	Stadion Slavija	Novi Sad	2000	Prole
105	Stadion Čukarica	Beograd	3500	Čuka

Кликом на икону са приказом графика отвара се табеларни приказ броја одржаних утакмица током протекле календарске године на сваком од стадиона. Уколико их има више од 10, приказати само првих 10 стадиона.



Потребно је базу проширити могућношћу памћења играча. За играча треба памтити име, презиме, датум рођења и из које земље је пореклом. Играч у различитим периодима може да игра за различите клубове, па се зато памти датум када је почeo и датум до када је играо за неки тим. Први датум је обавезан, а други је празан уколико играч још увек игра за тај тим. У самој бази треба обезбедити да други датум или није постављен или је мањи од првог датума.

Креирати дијаграм и бекап стања проширене базе.

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А13**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: ДВД колекција**

Позориште „Звезда“ има већ дуги низ година своју приватну ДВД колекцију, која поседује велики број домаћих и страних филмова разних продуцената. За преглед и вођење евиденције имају апликацију са базом података чија структура је дата на слици у прилогу. Приликом прегледа постојећих података уочили су да подаци о продуцентима нису ажурни. Постојећа апликација нема функционалност промене података па су због тога одлучили да своју апликацију прошире додатном опцијом и омогуће корекцију података. Такође се указала потреба да сваког момента могу да провере колико филмова у својој колекцији имају од сваке продуцентске куће. У циљу евиденције коришћења ДВД -а потребно је проширити постојећу базу са новим подацима.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_12**, где се као име и презиме уноси име ученика.

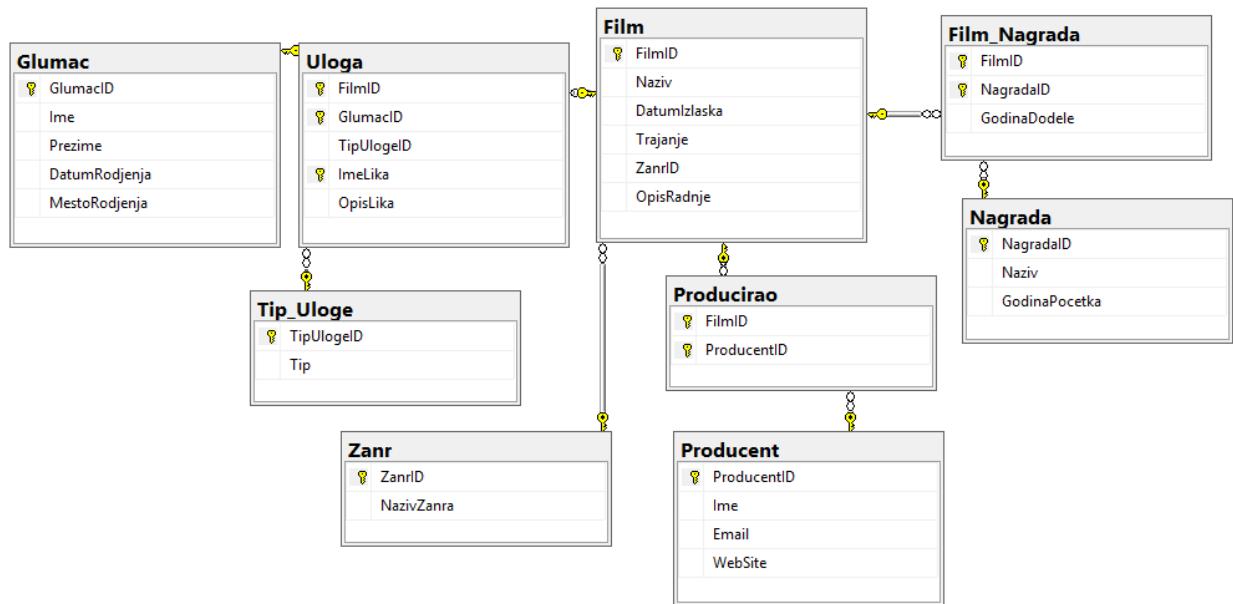
Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A12\Потребни ресурси**.

## Прилог за задатак:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати почетну форму са које се кликом на ставке менија приступа појединачним функционалностима апликације.

Креирана форма поред менија треба да приказује и податке о свим продуцентима у **listBox** контроли (шифра, назив продуцентске куће). Такође на форму треба поставити поља за појединачни приказ изабраног продуцента. Појединачна поља иницијално треба попунити са подацима продуцента који има најмању вредност шифре. Одабир продуцента, чије податке је потребно променити врши се селекцијом реда у **listBox** контроли. Селектоване податке приказати у појединачним пољима форме. Сва поља продуцента, сем шифре су подложна промени. Кликом да дугме менија **Izmeni** све вредности о приказаном продуценту се уписују у базу. Ажурирањем податка у бази освежити и приказ у **listBox** контроли. Поруку о успешности измене података приказати у **MessageBox**-у. Изглед форме је дат на слици која у заглављу има назив **Korekcija podataka - Producentska kuća**.

Функционалност менија:

- Ставка менија **Izmeni** омогућава ажурирање података о продуценту;
- Ставка менија **Analiza** приказује форму са графиконом;
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство;
- Ставка менија **Izlaz** затвара апликацију.

Korekcija podataka - Producentska kuća

Izmeni

Šifra	<input type="text" value="8"/>
Ime	<input type="text" value="BBC Films"/>
E-mail	<input type="text" value="adresa@gmail.com"/>

Analiza

O aplikaciji

Šifra	Naziv producentske kuće	E-mail adresa
9	20th Century Fox	filmwjtk@outlook.net
17	American International Pictures	skrcpn72@hotmail.org
8	BBC Films	
18	Capitol Films	gvymt90@gmail.org
19	Cinemation Industries	pzqim67@outlook.org
20	Columbia TriStar Motion Picture Group	bzmcm5@hotmail.net
6	DreamWorks	xfsojp18@outlook.org
4	Four Star Television	ummi7@mail.org

Korekcija podataka - Producentska kuća

Izmeni

Šifra	<input type="text" value="8"/>
Ime	<input type="text" value="BBC Films"/>
E-mail	<input type="text" value="adresa@gmail.com"/>

Uspesna izmena  
OK

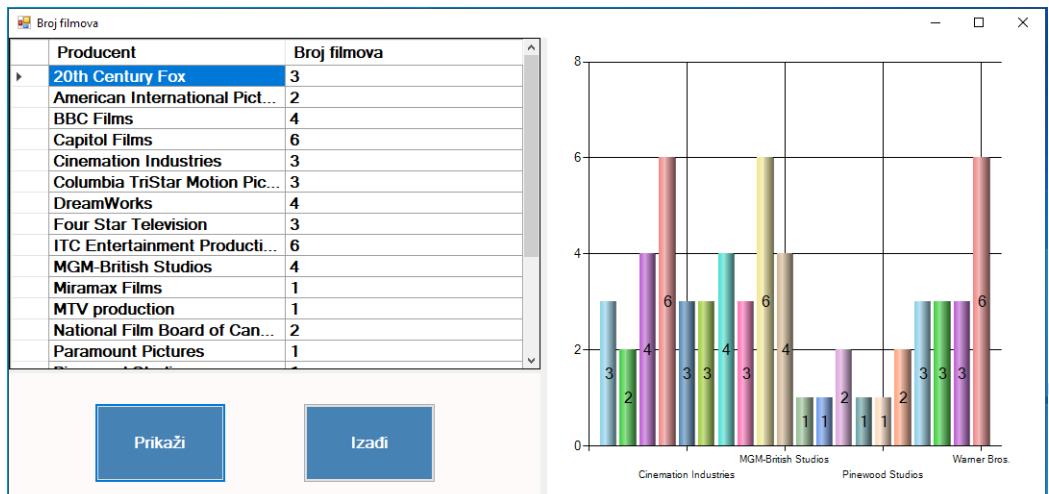
Analiza

O aplikaciji

Šifra	Naziv producentske kuće	E-mail adresa
9	20th Century Fox	filmwjtk@outlook.net
17	American International Pictures	skrcpn72@hotmail.org
8	BBC Films	adresa@gmail.com
18	Capitol Films	gvymt90@gmail.org
19	Cinemation Industries	pzqim67@outlook.org
20	Columbia TriStar Motion Picture Group	bzmcm5@hotmail.net
6	DreamWorks	xfsojp18@outlook.org
4	Four Star Television	ummi7@mail.org

Ставка менија **Analiza**, графички приказује укупан број филмова сваког продуцента. Приказ реализовати на новој форми. На форми се налазе два дугмата. Дугме **Prikaži** попуњава табелу и исцртава график а дугме **Izađi** затвара отворен образац и враћа на почетну форму.

Образац треба да изгледа као на слици:



Осмислите проширење базе којим се омогућава чување података о посуђивању филмова од стране запослених (име, презиме, телефон, адреса), као и податак о томе да ли је филм неко узео или је слободан. Потребно је водити рачуна да један исти филм запослени може узимати више пута. Такође је потребно табелу **Glumac** проширити са польем надимак глумца.

Креирати дијаграм и бекап стања проширене базе.

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А14

НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Дистрибуција лекова

"АПОТЕКА РЕГИОН" је једна од најбрже растућих компанија у области развоја тржишта фармацеутских производа у Србији. За преглед и вођење евидентије имају апликацију са базом података чија структура је дата на слици у прилогу. Приликом прегледа постојећих података уочили су да су неки од лекова које имају у понуди избачени из листе дозвољених лекова и да их је потребно избрисати из понуде за продају. Такође се указала потреба да се омогући приказ произвођача по броју лекова у циљу добијања информације који произвођачи су тренутно најзаступљенији на тржишту. У циљу евидентије набавке потребно је проширити постојећу базу са новим подацима.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_14**, где се као име и презиме уноси име кандидата.

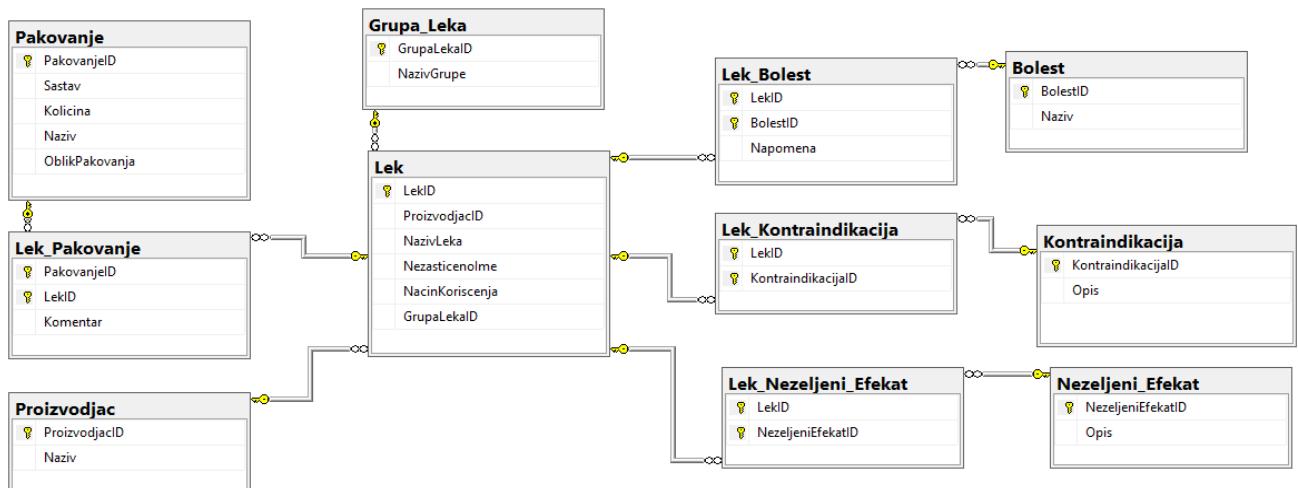
Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка кандидат може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A14\Потребни ресурси**.

## Прилог за задатак:

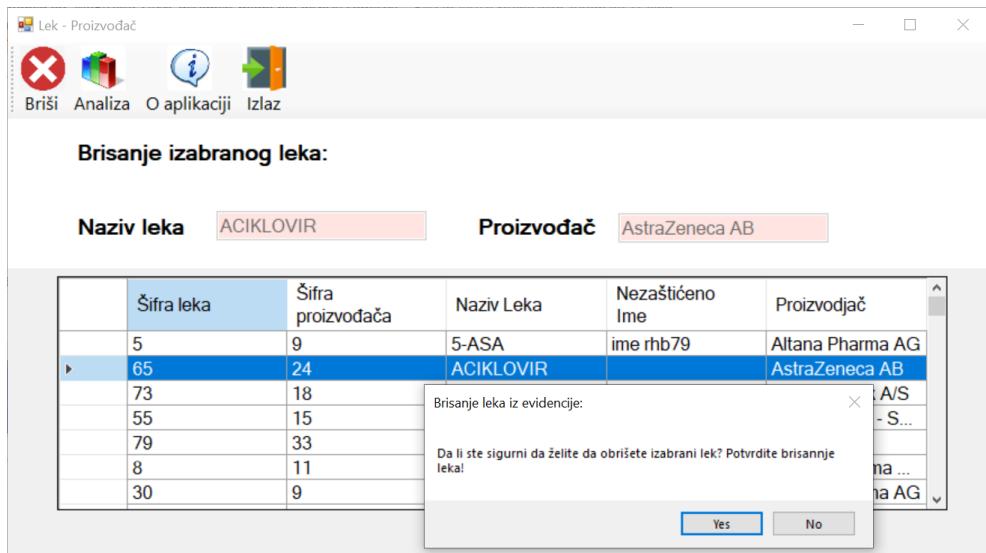


Развојним планом апликације предвиђено је да се уради следеће:

Креирана форма поред менија треба да приказује и податке о свим лековима у контроли **dataGridView** (шифра лека, шифра производићача, назив лека, незаштићено име, производићач). Такође на форму треба поставити поља за појединачни приказ лека и производићача. За приказ ова два поља користити **textBox** контролу. Корисник не може да уноси текст у **textBox** контролу. Појединачна поља иницијално треба попунити са подацима лека и његовог производићача који се налази у првој врсти табеле. Приказ у табели је сортиран по називу лека у растућем абецидном поретку. Након селекције реда у табели подаци о леку се приказују у појединачним пољима форме. Кликом да дугме менија **Briši**, из базе података се брише лек, чије су вредности приказане у појединачним пољима форме. Пре извршене акције брисања корисник треба да потврди да је сигуран да жели да брише одабран лек. Након брисања податка у бази освежити и приказ у **dataGridView** контроли. Изглед форме је дат на слици која у заглављу има назив **Lek - Proizvođač**. Поруку о успешности брисања приказати у **MessageBox**-у.

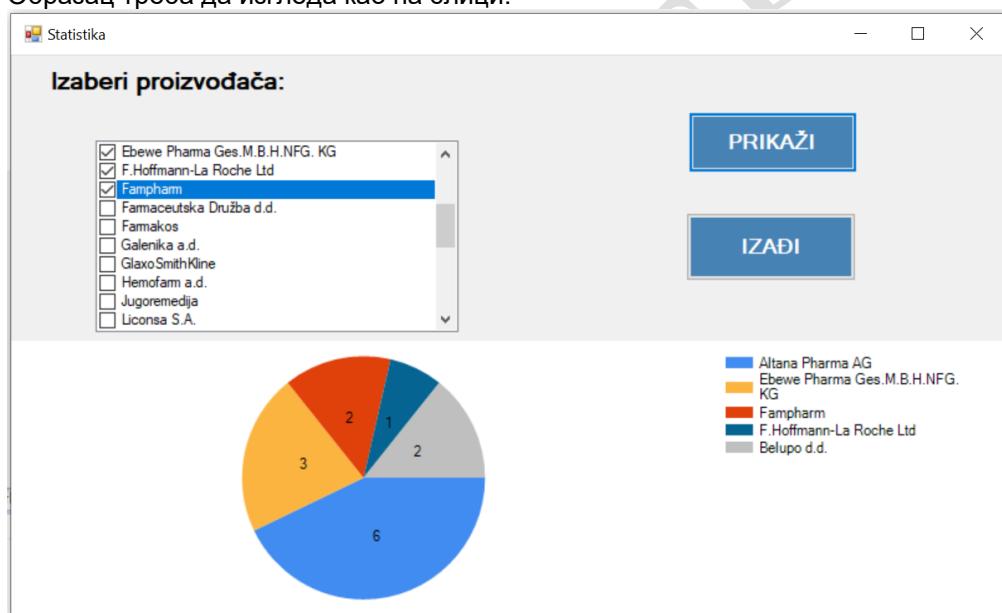
Функционалност менија:

- Ставка менија **Briši** омогућава брисање података о леку задатог производићача;
- Ставка менија **Analiza** приказује форму са графиконом;
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство;
- Ставка менија **Izlaz** затвара апликацију.



Ставка менија **Analiza**, графички приказује укупан број лекова сваког производјача. Приказ реализовати на новој форми. На форми се налазе два дугмента. Дугме **PRIKAŽI** исцртава график за чекиране производјаче а дугме **IZAĐI** затвара отворен образац и враћа на почетну форму.

Образац треба да изгледа као на слици:



Осмислити проширење базе којим се омогућава чување података о наруџбинама лекова од стране апотека. Треба водити рачуна о томе да свака апотека може да наручи више лекова а један лек може да буде наручен од стране више апотека, такође и апотека може да наручи исти лек више пута. Наруџбина садржи и информацију о количини нарученних лекова. Апотека осим своје шифре има и име апотеке, број телефона, имејл-адресу, адресу, град и државу. Такође је потребно табелу **Pakovanje** проширити са пољем за евиденцију датума до ког је лек употребљив.

Креирати дијаграм и бекап стања проширене базе.

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А15

НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Клинички центар

Градска болница "Клинички центар" се бави истраживачким радом и одређивањем нових терапија за евидентиране болести. За преглед и вођење евидентије о болестима и лековима имају апликацију са базом података чија структура је дата на слици у прилогу. Потребно је проширити постојећу апликацију са додавањем функционалности уписа лека за изабрану болест као и за приказ укупног броја различитих лекова који припадају изабраним групама лекова. У циљу ефикаснијег одређивања нових терапија потребно је проширити постојећу базу са подацима који ће прецизније описати болест.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_15**, где се као име и презиме уноси име кандидата.

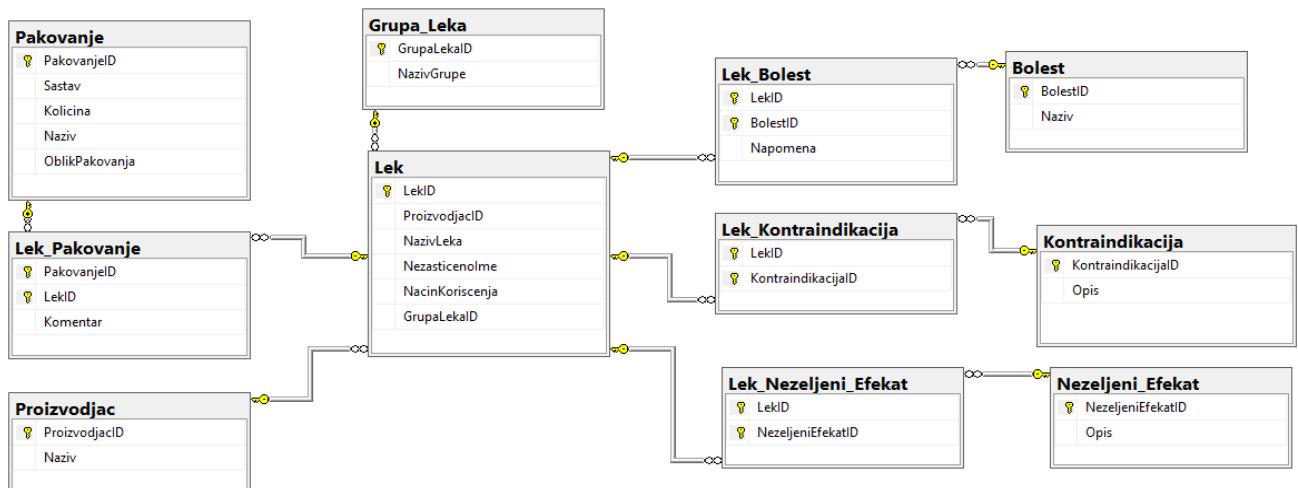
Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка кандидат може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A15\Потребни ресурси**.

## Прилог за задатак:



Развојним планом апликације предвиђено је да се уради следеће:

Креирана форма поред менија треба да приказује и податке о свим болестима у контроли **dataGridView** (болест, лек, производјач). Такође на форму треба поставити поља за појединачни приказ болести, лека и поља за напомену. За приказ поља за болест и лек користити **comboBox** контролу. Појединачна поља иницијално треба попунити са подацима болести и лека који се налази у првој врсти табеле. Приказ у табели је сортиран по називу болести у растућем алфабетском поретку. Након селекције реда у табели подаци о болети и леку се приказују у појединачним пољима форме. Кликом на дугме **UPIŠI**, у базу података се уписују вредности које су селектоване у контролама за приказ болести и лека. Након покушаја уписа селектоване комбинације, корисник треба да добије поруку о успешности уписа. Изглед форме је дат на слици која у заглављу има назив **Lek - Bolest**. У контроли за табеларни приказ података приказују се подаци који постоје у бази након уписа. Поруку о успешности уписа приказати у **MessageBox**-у.

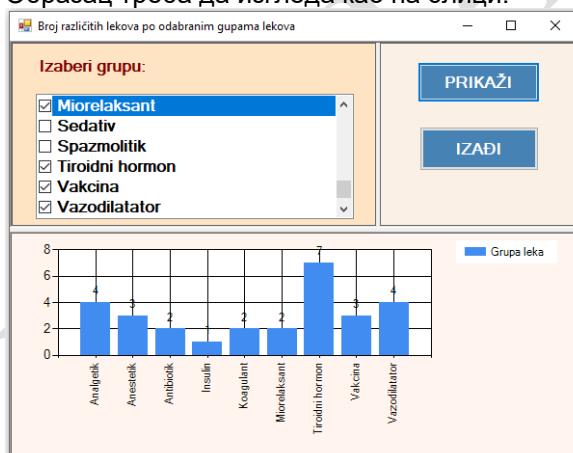
Функционалност менија:

- Ставка менија **Upiši** омогућава упис лека за изабрану болест;
- Ставка менија **Analiza** приказује форму са графиконом;
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство;
- Ставка менија **Izlaz** затвара апликацију.

Приказ након уписа комбинације болест – лек у базу:

Ставка менија **ANALIZA**, графички приказује укупан број различитих лекова који припадају изабраним групама лекова. Приказ реализовати на новој форми. На форми се налазе два дугмента. Дугме **PRIKAŽI** исцртава график за чекиране групе лекова а дугме **IZADI** затвара отворен образац и враћа на почетну форму.

Образац треба да изгледа као на слици:



Осмислити проширење базе којим се омогућава чување података о симптомима болести. Свака болест може да има више од једног симптома а један симптон може да буде карактеристика више болести. Симптом болести описан је са следећим пољима: шифра, опис и јачина. Јачина симптома се испољава у три нивоа: благ, изражен и јак. Такође је потребно табелу **Proizvodjac** проширити са пољем адреса и град.

Креирати дијаграм и бекап стања проширене базе.

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А16**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Изложба паса**

Кинолошки савез Републике Србије промовише кинологију, едукује одгајиваче, промовише савестан узгој и води бригу о одржавању аутохтоних српских пасмина. КСС је такође стални члан Међународне кинолошке федерације. Савез окупља бројна кинолошка друштва из целе Србије, промовише нашу кинологију кроз добру организацију кинолошких манифестација (међународних, националних изложби, специјализованих изложби, утакмица и др.), врши сталну едукацију Кинолошких судија, омогућава подизање квалитета узгоја свих раса и прецизно води матичну евиденција паса. За преглед и вођење евиденције о изложбама имају апликацију са базом података чија структура је дата на слици у прилогу. Потребно је проширити постојећу апликацију са додавањем функционалности пријаве пса на заказану изложбу и са извештајем о броју учесника на изложбама по категоријама. У циљу праћења порекла паса, који учествују на изложбама потребно је проширити постојећу базу са подацима о одгајивачницама.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_16**, где се као име и презиме уноси име кандидата.

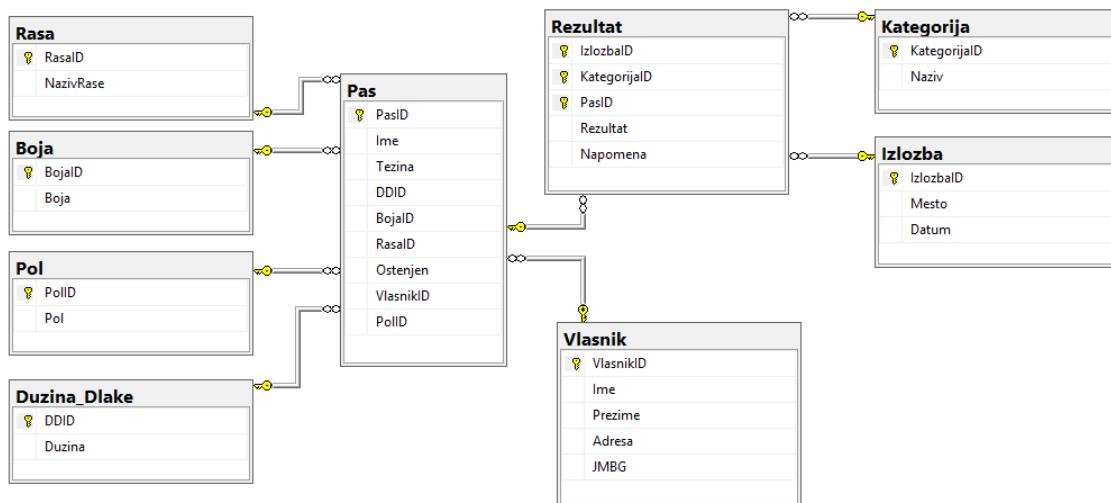
Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка кандидат може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A16\ Потребни ресурси**.

## Прилог за задатак:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која има четири таба. Пријава пса на такмичење се реализације помоћу таба **Unesi prijavu**. Садржај овог таба је образац са пољима која треба попунити. Приказ обрасца је дат на доњој слици која у заглављу форме име назив „**Prijava na izložbu**“. За унос потребних података приликом пријаве користити **comboBox** компоненте.

**ComboBox** за унос пса који се такмичи треба попунити са шифром и називом пса. Подаци у **comboBox**-у су сортирани по растућем редоследу шифре.

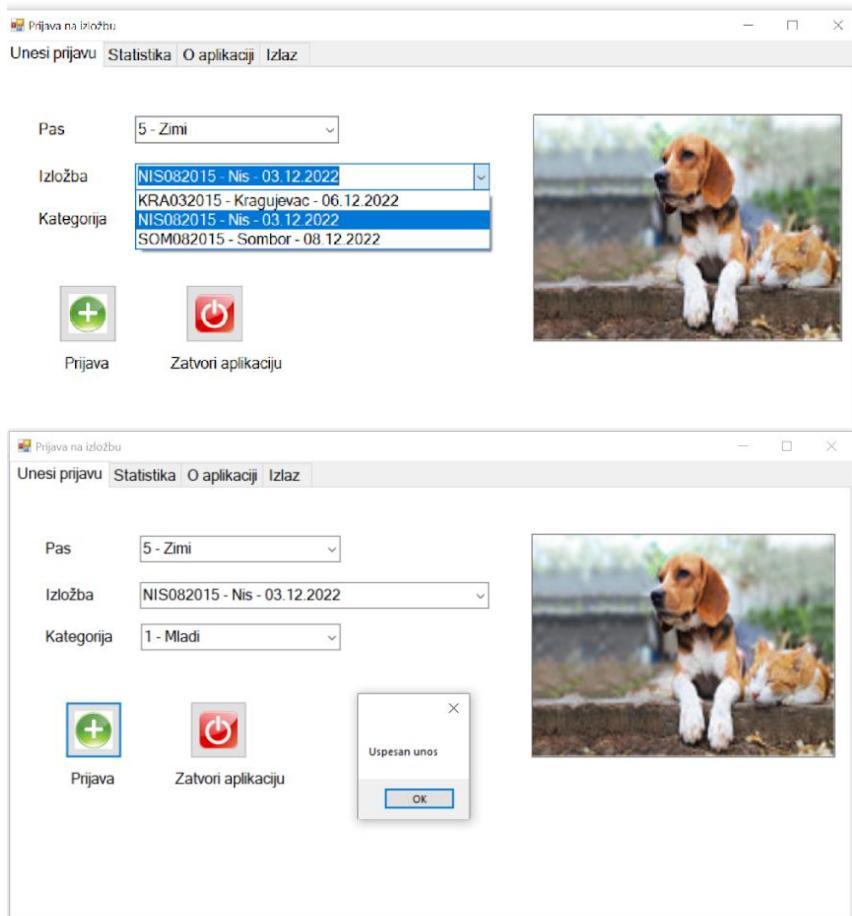
**ComboBox** за унос изложбе треба попунити са шифром и местом изложбе. Приказани подаци су сортирани по растућем редоследу шифре. Обезбедити да се пас може пријавити најкасније 48 часова пре почетка изложбе.

**ComboBox** за унос категорије треба попунити са шифром и називом категорије. Приказани подаци су сортирани по растућем редоследу шифре.

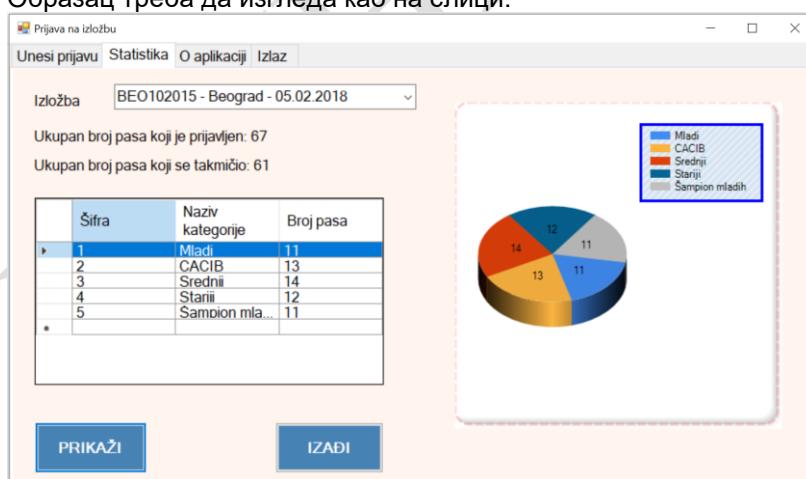
Селекцијом пса, изложбе и категорије врши се одабир комбинације, односно попуњава се пријава за учешће на изложби. Након клика на дугме **Prijava** евидентира се и памти попуњена пријава. Приликом покушаја уписа пријаве проверава се да ли је изабран пас већ пријављен на селектовану изложбу. У случају да је пријављен, корисник добија обавештење о томе. Такође приказати поруку о успешности уписа у **MessageBox**-у. Дугме **Zatvoriti aplikaciju** затвара апликацију.

Функционалност таб менија:

- Ставка менија **Unesi prijavu** омогућава попуњавање пријаве;
- Ставка менија **Statistika** приказује форму са графиконом;
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство;
- Ставка менија **Izlaz** затвара апликацију.



Други таб форме **Statistika**, табеларно и графички приказује колико се такмичило паса у свакој категорији за одређену изложбу. У **ComboBox** изложба треба приказати само изложбе које су се одржале. Други таб форме организовати као на слици. На другом табу се налазе два дугмета. Дугме **PRIKAŽI** исцртава график за селектовану изложбу а дугме **IZAĐI** затвара отворен образац и враћа кориснику на први таб форме. Такође, на другом табу треба исписати укупан број паса који је пријављен и укупан број паса који се такмичио.



Осмислiti проширење базе којим се омогућава чување података о одгајивачницама паса. Пас може имати само једног одгајивача, а одгајивач може имати више паса. За одгајивача је потребно чувати поред података о имени, број телефона, имејл-адресу, адресу и државу. У табели раса треба додати и напомену у којој ће се чувати додатна запажања о карактеристикама расе.

Креирати дијаграм и бекап стања проширене базе.

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А17

НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Евиденција возила

Авто плац "Превоз" се бави изнајмљивањем возила. За потребе ауто плаца креирана је база података у којој се води евиденција о свим возилима. Потребно је проширити постојећу апликацију са додавањем функционалности ажурирања података о аутомобилима, као и за приказ просечне цене аутомобила по години производње за задати модел. У циљу повећања броја изнајмљених аутомобила потребно је проширити постојећу базу са подацима о сајтовима за њихово рекламирање. Креирати десктоп апликацију која ће омогућити рад са постојећом базом.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_17**, где се као име и презиме уноси име ученика.

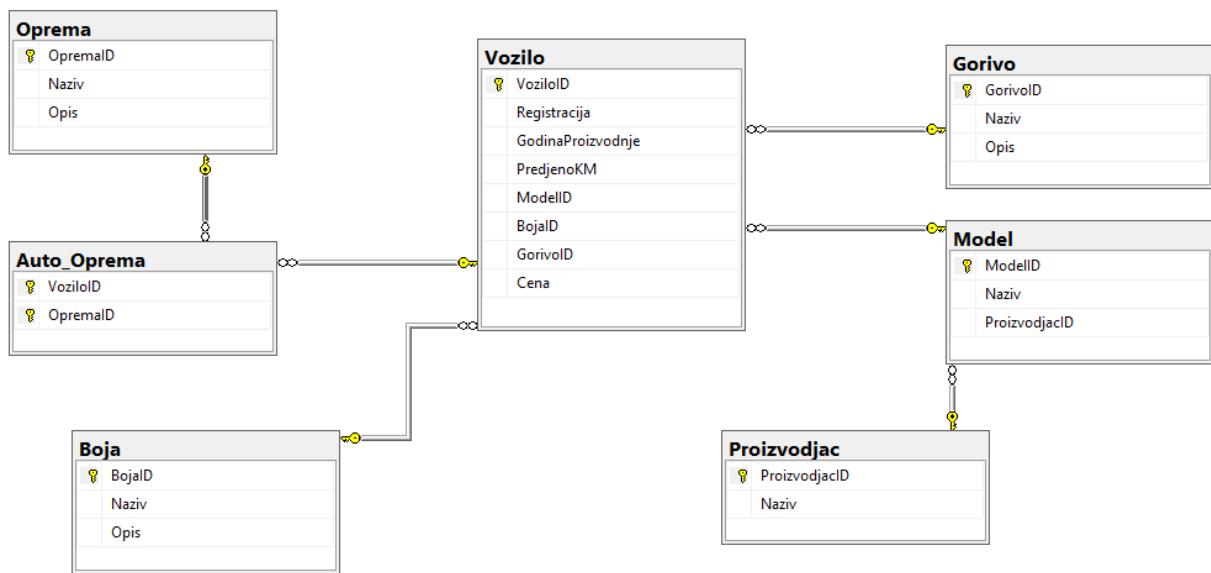
Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A17\Потребни ресурси**.

## Прилог за задатак:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која податке о свим аутомобилима приказује у **ListBox** контроли. Приказани подаци треба да буду сортирани по шифри. Након селекције реда у **ListBox** контроли или променом шифре у пољу шифра на форми, подаци о аутомобилу се приказују у одговарајућим пољима форме. Омогућити кориснику измену селектованог аутомобила. Кликом на дугме **Priprema za izmenu** поље за шифру има дозволу само за читање и појављује се дугме **Odustani**. Кликом на дугме **Odustani** форма се враћа у почетно стање. Након измене ажурирати садржај **ListBox** контроле. Поруку о успешности уноса приказати у **MessageBox**-у.

Функционалност менија:

- Ставка менија **Izmeni** омогућава ажурирање података изабраног аутомобила;
- Ставка менија **Analiza** приказује форму са графиконом;
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство;
- Ставка менија **Izlaz** затвара апликацију.

Pregled vozila u garaži

Izmeni Analiza O aplikaciji Izlaz

<b>Opšti podaci:</b>	<b>Dodatačni podaci:</b>
Šifra: <input type="text" value="1"/>	Model: <input type="text" value="Passat"/>
Registracija: <input type="text" value="WP-069-VG"/>	Boja: <input type="text" value="Bela"/>
Godina proizvodnje: <input type="text" value="2006"/>	Gorivo: <input type="text" value="Evrodizel"/>
Predjeno kilometara: <input type="text" value="22.394,88"/>	
Cena: <input type="text" value="15.072,38"/>	

**Priprema za izmenu**

Vozilo ID	Registracija	Godina proizvodnje	Predjeno KM	ModelID	BojaID	GorivoID	Cena
1	WP-069-VG	2006	22.394,88	81	9	1	15.072,38
2	MF-308-CS	1998	138.470,31	121	10	2	16.711,86
3	XS-403-XH	2010	134.160,05	113	2	5	15.824,41
4	IE-388-UA	2002	177.556,65	83	11	1	12.046,90
5	AK-752-VF	1995	146.662,65	31	3	5	5.708,60
6	DW-759-VV	2006	122.610,74	83	2	3	12.059,93
7	LE-836-YZ	2008	33.164,14	134	11	6	18.308,49
8	WT-238-FP	2011	88.860,59	66	0	1	9.959,20

Pregled vozila u garaži

Izmeni Analiza O aplikaciji Izlaz

<b>Opšti podaci:</b>	<b>Dodatni podaci:</b>
Šifra <input type="text" value="1"/>	Model <input type="text" value="Passat"/>
Registracija <input type="text" value="WP-069-VG"/>	Boja <input type="text" value="Bela"/>
Godina proizvodnje <input type="text" value="2006"/>	Gorivo <input type="text" value="Evrodizel"/>
Predjeno kilometara <input type="text" value="22.394,88"/>	
Cena <input type="text" value="15.072,38"/>	

**Vozilo ID Registracija Godina proizvodnje Predeno KM ModelID BojaID GorivoID Cena**

1	WP-069-VG	2006	22.394,88	81	9	1	15.072,38
2	MF-308-CS	1998	138.470,31	121	10	2	16.711,86
3	XS-403-XH	2010	134.160,05	113	2	5	15.824,41
4	IE-388-UA	2002	177.556,65	83	11	1	12.246,90
5	AR-752-VF	1995	146.662,65	31	3	5	5.708,60
6	DW-759-VV	2006	122.610,74	83	2	3	12.059,93
7	LE-836-YZ	2008	33.164,14	134	11	6	18.308,49
8	WT-238-FP	2011	88.860,59	66	0	1	9.959,20

**Priprema za izmenu** **Odustani**

Pregled vozila u garaži

Izmeni Analiza O aplikaciji Izlaz

<b>Opšti podaci:</b>	<b>Dodatni podaci:</b>
Šifra <input type="text" value="1"/>	Model <input type="text" value="Passat"/>
Registracija <input type="text" value="WP-069-VG"/>	Boja <input type="text" value="Žuta"/>
Godina proizvodnje <input type="text" value="2006"/>	Gorivo <input type="text" value="Evrodizel"/>
Predjeno kilometara <input type="text" value="22.394,88"/>	
Cena <input type="text" value="15.072,38"/>	

**Vozilo ID Registracija Godina proizvodnje Predeno KM ModelID BojaID GorivoID Cena**

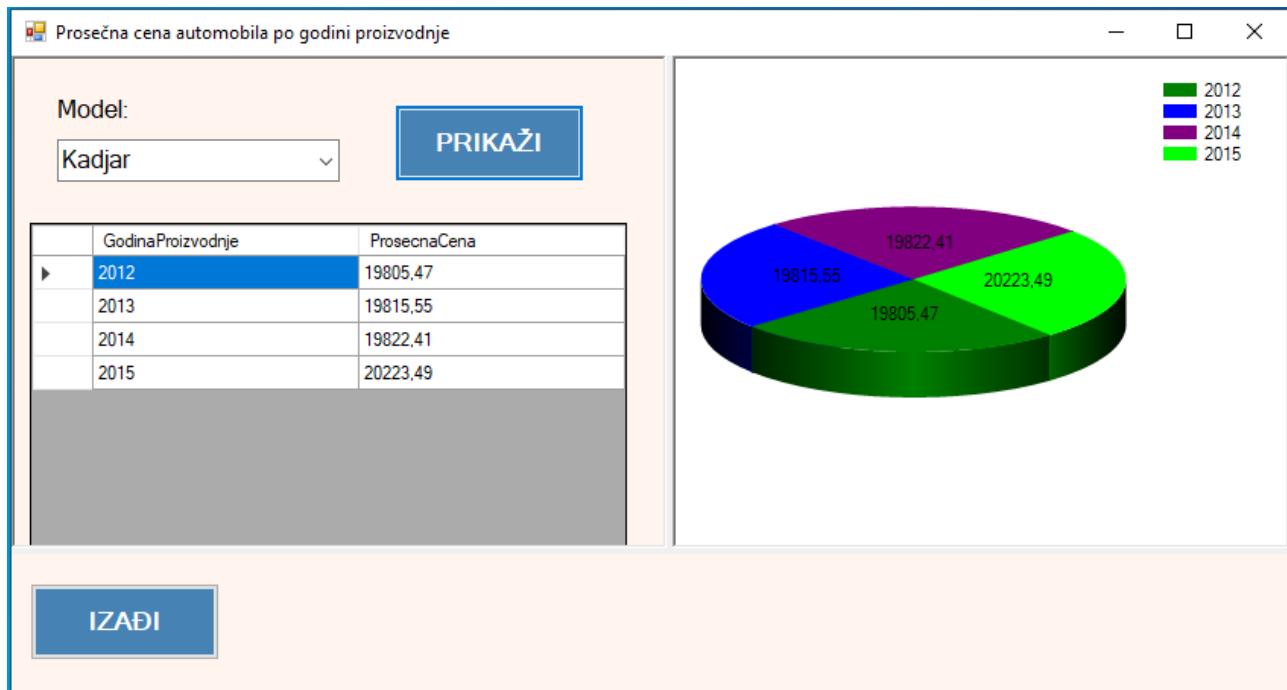
1	WP-069-VG	2006	22.394,88	81	9	1	15.072,38
2	MF-308-CS	1998	138.470,31	121	10	2	16.711,86
3	XS-403-XH	2010	134.160,05	113	2	5	15.824,41
4	IE-388-UA	2002	177.556,65	83	11	1	12.246,90
5	AR-752-VF	1995	146.662,65	31	3	5	5.708,60
6	DW-759-VV	2006	122.610,74	83	2	3	12.059,93
7	LE-836-YZ	2008	33.164,14	134	11	6	18.308,49
8	WT-238-FP	2011	88.860,59	66	0	1	9.959,20

**Priprema za izmenu** **Odustani**

Ставка менија **Analiza**, табеларно и графички приказује просечну цену аутомобила по години производње за задати модел. Дугме **PRIKAŽI** исцртава график за селектован модел а дугме **IZADI** затвара отворен образац и враћа на почетну форму.

Образац треба да изгледа као на слици:





Осмислите проширење базе којим се омогућава чување података о сајту који се користи за рекламирање возила. Возила је потребно рекламирати на различитим сајтовима. Потребно је за сваки сајт чувати назив сајта и детаљан опис. Обзиром да различита возила могу бити рекламирана на различитим сајтовима везу више на више између ове две табеле трансформисати у две везе један на више увођењем нове табеле. Такође је потребно табелу *Proizvodjac* проширити са пољем опис.

Креирати дијаграм и бекап стања проширене базе.

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А18

НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Ауто плац - Гаража

Ауто плац "Гаража" се бави изнајмљивањем возила. За потребе ауто плаца креирана је база података у којој се води евиденција о свим возилима. Потребно је проширити постојећу апликацију са додавањем функционалности уписа података за нове аутомобиле, као и за приказ колико аутомобила за сваки модел се тренутно налази на плацу а да је њихова цена у задатом опсегу. У циљу повећања броја изнајмљених аутомобила потребно је проширити постојећу базу са подацима о сајтовима за њихово рекламирање. Креирати десктоп апликацију која ће омогућити рад са постојећом базом.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_18**, где се као име и презиме уноси име кандидата.

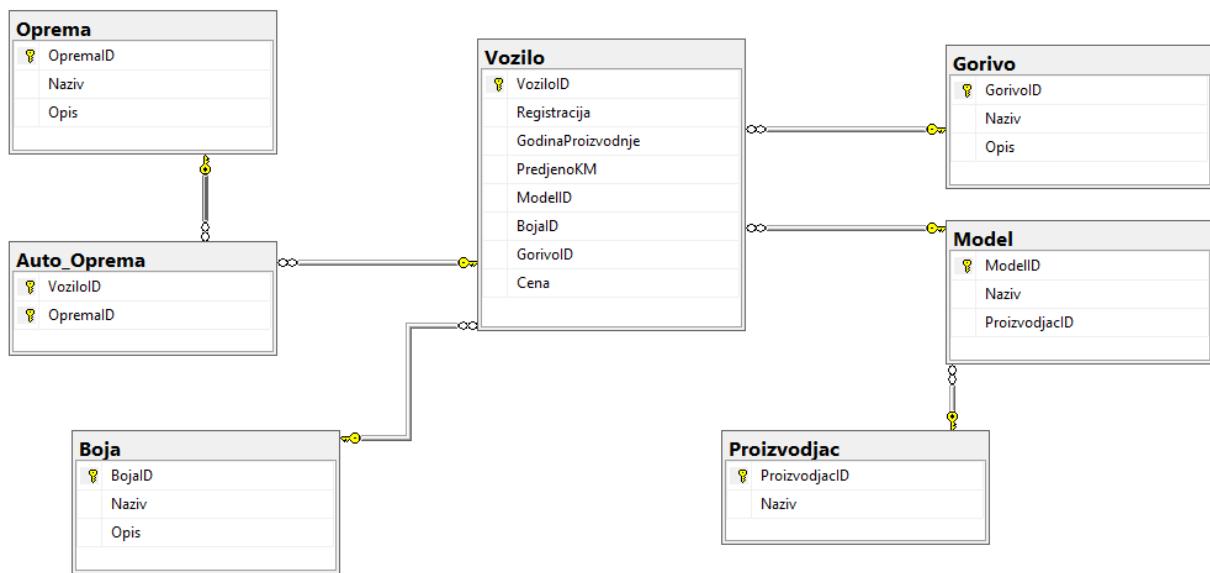
Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка кандидат може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A18\Потребни ресурси**.

## Прилог за задатак:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму на којој се врши приказ података о свим аутомобилима. За приказ тражених података употребити **listBox** контролу. Приказани подаци треба да буду сортирани по шифри. Такође на форму треба поставити поља за појединачни приказ потребних података о аутомобилу. За унос потребних података за модел, боју и гориво користити **comboBox** контроле. Након селекције реда у **listBox** контроли или променом шифре у пољу шифра на форми, подаци о аутомобилу се приказују у одговарајућим пољима форме. Омогућити кориснику унос новог аутомобила. Кликом на дугме **Priprema za upis** брише се садржај свих поља, поље за шифру има дозволу само за читање и појављује се дугме **Odustani**. Кликом на дугме **Odustani** форма се враћа у почетно стање. Приликом уписа аутомобила у базу аутоматски треба да се генерише вредност шифре. Након уноса ажурирати садржај **listBox** контроле. Поруку о успешности уноса приказати у **MessageBox**-у.

Функционалност менија:

- Ставка менија **Unesi** омогућава унос података новог аутомобила;
- Ставка менија **Analiza** приказује форму са графиконом;
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство;
- Ставка менија **Izlaz** затвара апликацију.

Pregled vozila u garaži

Unesi Analiza O aplikaciji Izlaz

<b>Opšti podaci:</b>	<b>Dodatačni podaci:</b>
Šifra <input type="text" value="1"/>	Model <input type="text" value="Jazz"/>
Registracija <input type="text" value="WP-069-VG"/>	Boja <input type="text" value="Bela"/>
Godina proizvodnje <input type="text" value="2006"/>	Gorivo <input type="text" value="Evrodizel"/>
Predeno kilometara <input type="text" value="22.394,88"/>	
Cena <input type="text" value="15.072,38"/>	

**Priprema za upis**

Vozilo ID	Registracija	Godina proizvodnje	Predeno KM	ModellID	BojaID	GorivoID	Cena
1	WP-069-VG	2006	22.394,88	107	9	1	15.072,38
2	MF-308-CS	1998	138.470,31	121	10	2	16.711,86
3	XS-403-XH	2010	134.160,05	113	2	5	15.824,41
4	IE-388-UA	2002	177.556,65	83	11	1	12.046,90
5	AR-752-VF	1995	146.662,65	31	3	5	5.708,60
6	DW-759-VV	2006	122.610,74	83	2	3	12.059,93
7	LE-836-YZ	2008	33.164,14	134	11	6	18.308,49
8	WT-238-FP	2011	88.860,59	66	0	1	9.959,20

Pregled vozila u garaži

Unesi Analiza O aplikaciji Izlaz

<b>Opšti podaci:</b>	<b>Dodatačni podaci:</b>
Šifra <input type="text"/>	Model <input type="text" value="Izaberi model..."/>
Registracija <input type="text"/>	Boja <input type="text" value="Izaberi boju..."/>
Godina proizvodnje <input type="text"/>	Gorivo <input type="text" value="Izaberi gorivo..."/>
Predeno kilometara <input type="text"/>	
Cena <input type="text"/>	

**Priprema za upis** **Odustani**

Vozilo ID	Registracija	Godina proizvodnje	Predeno KM	ModellID	BojaID	GorivoID	Cena
1	WP-069-VG	2006	22.394,88	107	9	1	15.072,38
2	MF-308-CS	1998	138.470,31	121	10	2	16.711,86
3	XS-403-XH	2010	134.160,05	113	2	5	15.824,41
4	IE-388-UA	2002	177.556,65	83	11	1	12.046,90
5	AR-752-VF	1995	146.662,65	31	3	5	5.708,60
6	DW-759-VV	2006	122.610,74	83	2	3	12.059,93
7	LE-836-YZ	2008	33.164,14	134	11	6	18.308,49
8	WT-238-FP	2011	88.860,59	66	0	1	9.959,20

Pregled vozila u garaži

Unesi Analiza O aplikaciji Izlaz

<b>Opšti podaci:</b>	<b>Dodatačni podaci:</b>
Šifra <input type="text"/>	Model <input type="text" value="Audi Q7"/>
Registracija <input type="text" value="NS-555-MM"/>	Boja <input type="text" value="Orange"/>
Godina proizvodnje <input type="text" value="2021"/>	Gorivo <input type="text" value="Evrodizel"/>
Predeno kilometara <input type="text" value="12000"/>	
Cena <input type="text" value="49000"/>	

Vozilo je ubaceno u bazu.  
 Šifra vozila je: 2003

**Priprema za upis** **Odustani**

Vozilo ID	Registracija	Godina proizvodnje	Predeno KM	ModellID	BojaID	GorivoID	Cena
1	WP-069-VG	2006	22.394,88	107	9	1	15.072,38
2	MF-308-CS	1998	138.470,31	121	10	2	16.711,86
3	XS-403-XH	2010	134.160,05	113	2	5	15.824,41
4	IE-388-UA	2002	177.556,65	83	11	1	12.046,90
5	AR-752-VF	1995	146.662,65	31	3	5	5.708,60
6	DW-759-VV	2006	122.610,74	83	2	3	12.059,93
7	LE-836-YZ	2008	33.164,14	134	11	6	18.308,49
8	WT-238-FP	2011	88.860,59	66	0	1	9.959,20

Pregled vozila u garaži

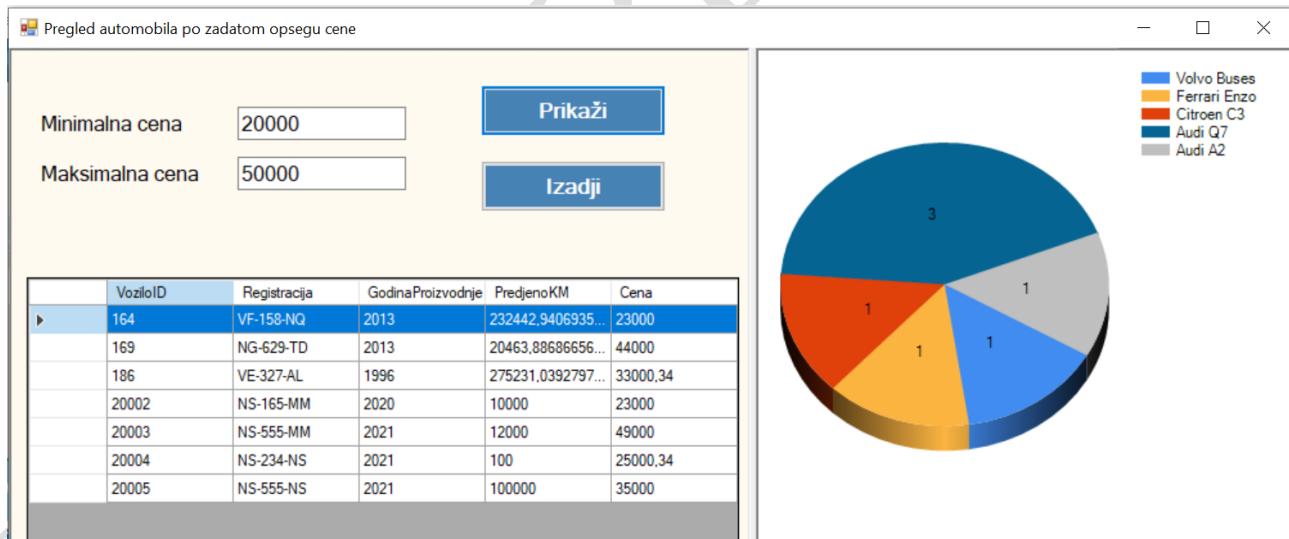
Unesi Analiza O aplikaciji Izlaz

<b>Opšti podaci:</b>	<b>Dodatajni podaci:</b>
Šifra <input type="text" value="20003"/>	Model <input type="text" value="Audi Q7"/>
Registracija <input type="text" value="NS-555-MM"/>	Boja <input type="text" value="Orange"/>
Godina proizvodnje <input type="text" value="2021"/>	Gorivo <input type="text" value="Evrodiesel"/>
Predjeno kilometara <input type="text" value="12.000,00"/>	
Cena <input type="text" value="49.000,00"/>	
<b>Priprema za upis</b>	

Vozilo ID	Registracija	Godina proizvodnje	Predjeno KM	ModellID	BojalD	GorivolD	Cena
19996	GP-002-LF	1999	219.879,39	32	10	3	5.819,98
19997	UK-723-EP	1996	169.143,58	134	2	1	18.327,65
19998	PA-625-AY	1997	151.969,14	140	5	6	19.064,84
19999	ZQ-625-OV	2004	39.851,96	47	7	2	7.645,93
20000	PH-764-LL	2008	123.315,55	44	11	6	7.321,20
20001	WW	12	12,00	135	1	5	11,00
20002	NS-165-MM	2020	10.000,00	135	1	1	23.000,00
<b>20003</b>	<b>NS-555-MM</b>	<b>2021</b>	<b>12.000,00</b>	<b>135</b>	<b>5</b>	<b>1</b>	<b>49.000,00</b>

Ставка менија **Analiza**, табеларно приказује све податке о аутомобилима чија је цена у опсегу унете минималне и максималне цене. Дугме **Prikaži** исцртава график који приказује број аутомобила за све моделе чија је цена у задатом опсегу. Дугме **Izadjii** затвара отворен прозор и враћам контролу на почетну страницу апликације.

Образац треба да изгледа као на слици:



Осмислити проширење базе којим се омогућава чување података о сајту који се користи за рекламирање возила. Возила је потребно рекламирати на различитим сајтовима. Потребно је за сваки сајт чувати назив сајта и детаљан опис. Обзиром да различита возила могу бити рекламирана на различитим сајтовима везу више на више између ове две табеле трансформисати у две везе један на више увођењем нове табеле. Такође је потребно табелу **Proizvodjac** проширити са пользима адреса, град и опис.

Креирати дијаграм и бекап стања проширене базе.

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А19

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Евиденција запослених**

Радна организација чува податке о својој организационој структури: податке о секторима, руководиоцима сектора, као и о свим запосленим радницима у секторима фирме. Сваки сектор има свог управника. Структура базе је дата на слици у прилогу. Управни одбор је исказао потребу да се у бази памти целокупна историја управљања секторима, тј податке о свим претходним као и о тренутно активном руководиоцу за сваки од сектора. За руководиоце чији је мандат истекао, памтити датум постављања и датум разрешења са дужности, док се за оне који у још увек на дужности руководиоца памти само датум именовања. Именовањем новог управника, стари се разрешава дужности (тј, датум именовања новог је уједно и датум разрешења дотадашњег управника).

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

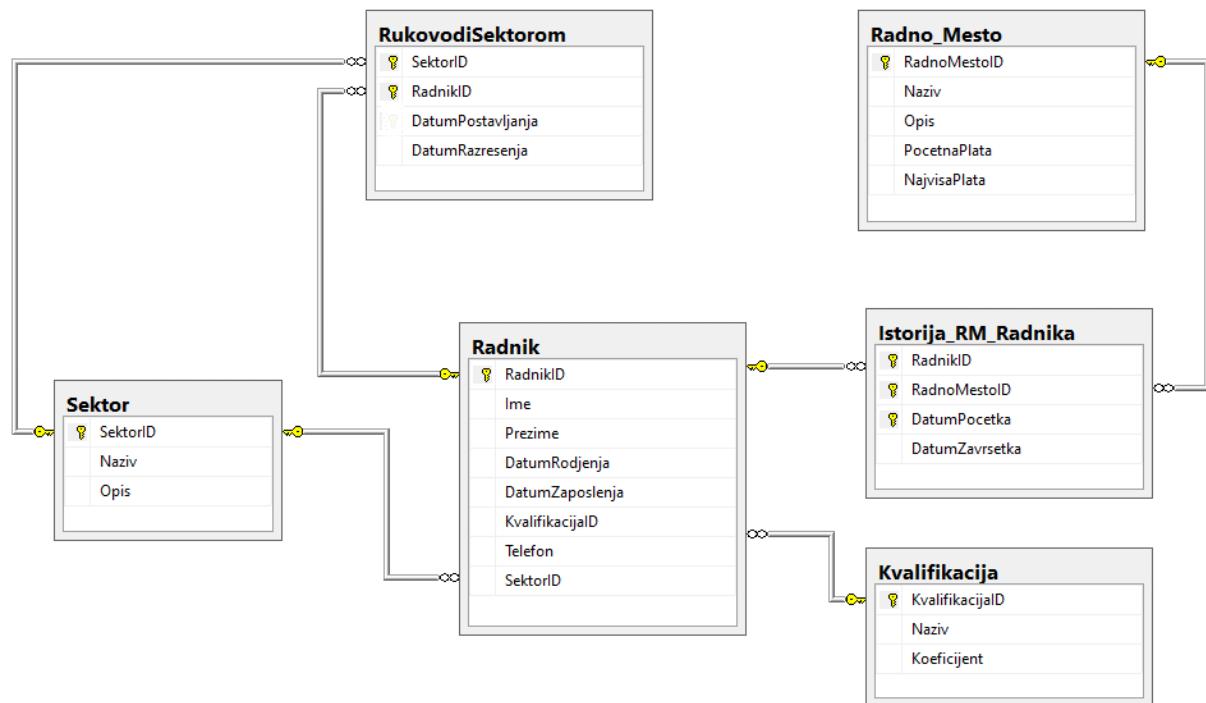
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A191**.

Резервна копија почетног стања базе дат је као **Прилог\_4ЕИТ\_A192**.

Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

**ПРИЛОГ:**



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која податке о свим управницима изабраног одељења приказује у *ListView* контроли (име презиме, кад је започео и кад је завршио мандат), а на предвиђену лабелу приказује назив селектованог сектора и податке о тренутном управнику

РАДНИК

The screenshot shows a Windows application window titled 'Sektori'. On the left, there's a section labeled 'Sektor' with a dropdown menu set to 'Sektor finansijskih poslova'. Below it is a section 'Podaci o sektoru' containing the text: 'Sektor finansijskih poslova rukovodilac: Balša Danilović postavljen: 07.07.2020'. At the bottom of this panel is a blue-outlined button labeled 'Promena rukovodioca'. To the right of this panel is a table showing staff information:

Sifra	Ime	Prezime	Od	Do
110	Lidija	Apostolović	20.12.2015	01.01.2017
136	Jadranka	Branković	01.01.2017	05.06.2018
206	Tina	Branković	05.06.2018	10.10.2018
114	Tatjana	Erceg	10.10.2018	07.07.2021
184	Balša	Danilović	07.07.2020	Do danas

A secondary window titled 'Novi rukovodilac' is open over the main window. It contains a label 'Rukovodilac i datum stupanja na dužnost:' followed by a dropdown menu. The dropdown menu lists various names and their IDs, with '(16) Veljko David' currently selected. The dropdown menu has a scroll bar.

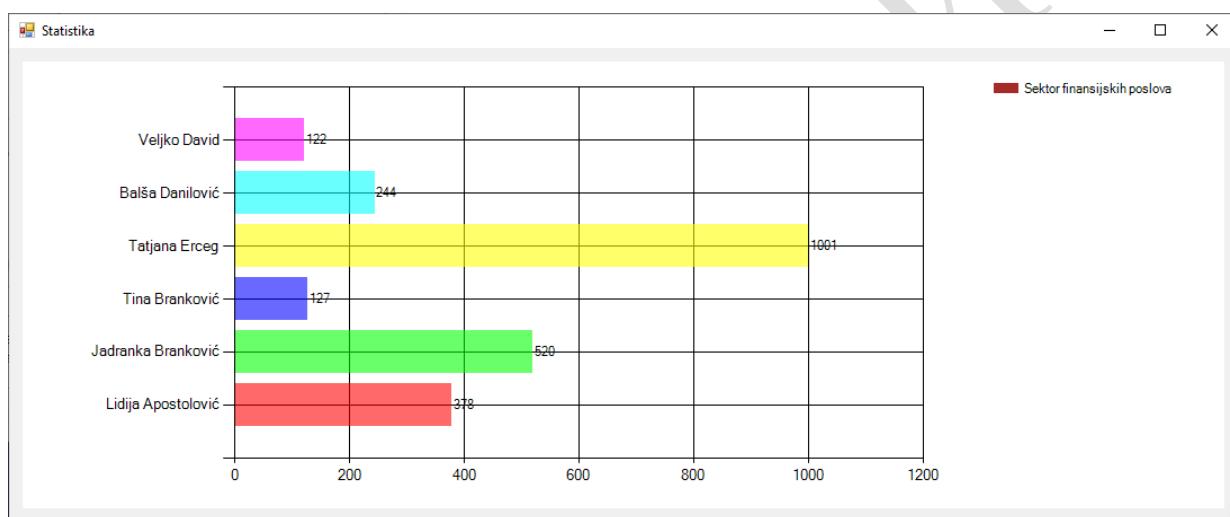
Када је изабран неки од постојећих сектора, укључује се дугме које омогућава промену руководиоца сектора. Кликом на дугме **Промена руководиоца**, отвара се форма на којој треба изабрати новог руководиоца и унети датум његовог именовања. **Обезбедити да се као руководилац неког сектора може изабрати само радник који ради у том сектору** (нови руководилац се бира из **comboBox** контроле, а датум се уноси путем текст поља са маском).

Треба омогућити да се именовањем новог управника, аутоматски, са истим датумом, разрешава дужности дотадашњи руководилац, а приказ на форми се освежава и приказују се актуелни-измењени подаци.

The screenshot shows the 'Novi rukovodilac' dialog box. It contains a label 'Rukovodilac i datum stupanja na dužnost:' above a dropdown menu. The dropdown menu is populated with names and IDs, with '(16) Veljko David' selected. Below the dropdown is a text input field containing the date '08.03.2021'. At the bottom of the dialog is a button labeled 'Unesi izmenu >>'

Sifra	Ime	Prezime	Od	Do
110	Lidija	Apostolović	20.12.2015	01.01.2017
136	Jadranka	Branković	01.01.2017	05.06.2018
206	Tina	Branković	05.06.2018	10.10.2018
114	Tatjana	Erceg	10.10.2018	07.07.2021
184	Balša	Danilović	07.07.2020	08.03.2021
16	Veljko	David	08.03.2021	Do danas

Дугме са сликом графика постаје активно када је селектован неки од сектора и кликом на њега, отвара се друга форма на којој је приказана статистика руковођења изабраног сектора – тј на графику су приказани сви руководиоци и дужина њиховог ангажовања на тој позицији, изражена у данима.



Структура постојеће базе дозвољава да један човек само у једном мандату буде управник неког сектора. Направити потребне измене у табели која чува податке о руководиоцима, тако да буде могуће да се један исти човек у више мандата постави на руководећу позицију сектора (за јединствену идентификацију мандата узети у обзир почетак мандата). Додати и ограничење базе које обезбеђује да датум почетка мандата мора бити мањи од датума завршетка мандата.

Осмислите проширење базе којим се омогућава чување података о локацијама на којима се налазе канцеларије појединих сектора. Једна канцеларија припада само једном сектору, али сектор може имати на располагању већи број канцеларија. За сваку локацију памтити адресу и капацитет, тј број радних места.

Креирати дијаграм и бекап стања проширене базе.

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–A20

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Туристичка агенција**

Туристичка агенција располаже базом у којој се региструју подаци о свим склопљеним уговорима продаје туристичких аранжмана. Агенција жали да омогући својим клијентима плаћање на рате у складу са склопљеним уговором. Стога је апликацији потребно додати део који омогућава уплате рата за аранжман и праћење преосталог дуга.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

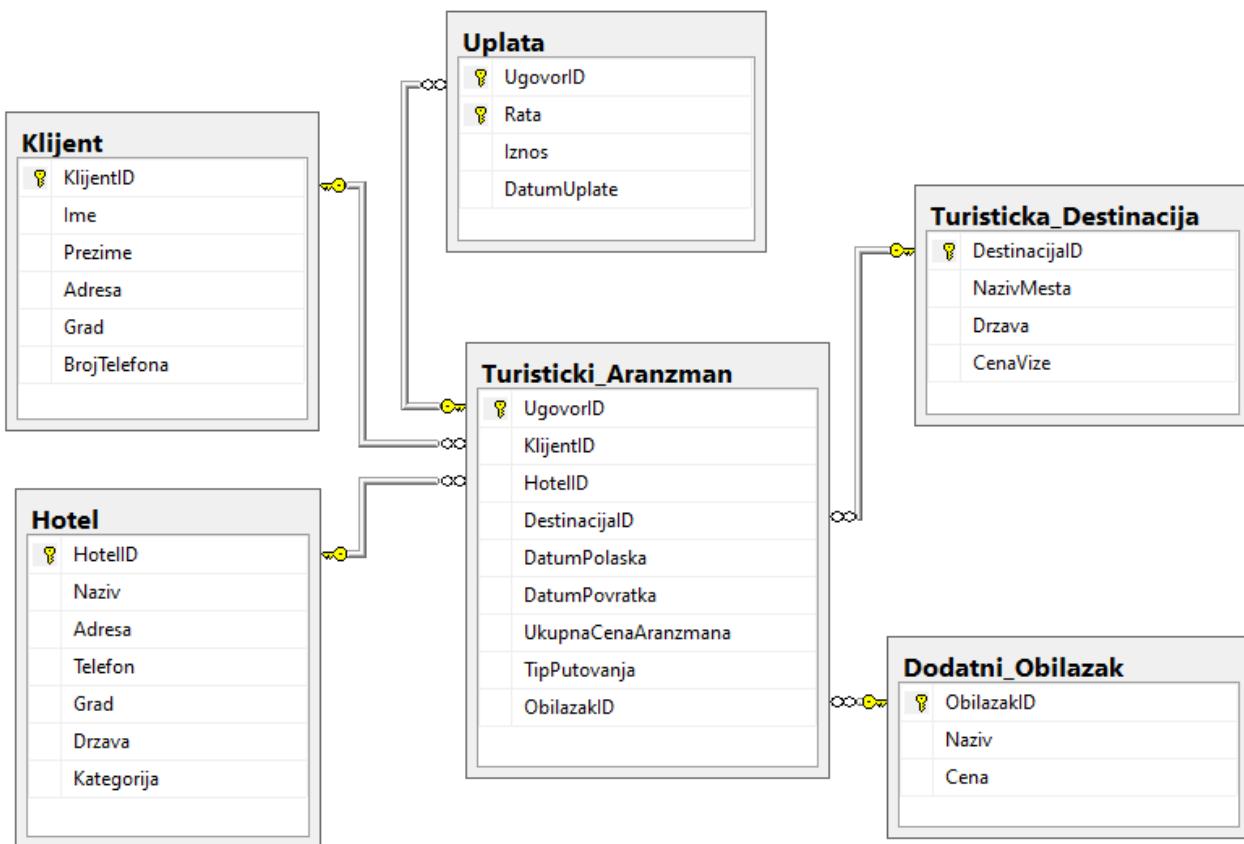
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A201**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A202**.

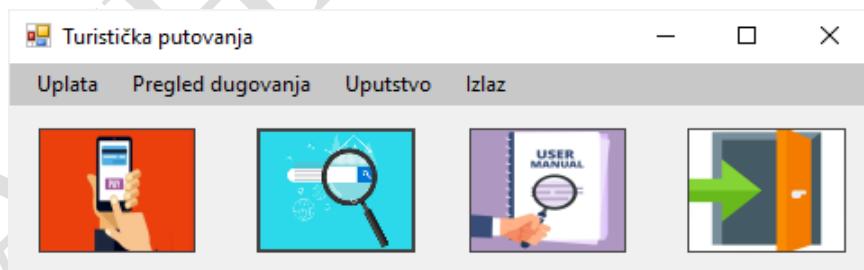
Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати почетну форму са које се било кликом на одговарајуће дугме или кликом на ставку менија приступа појединачним функционалностима апликације.



Прво дугме **Уплата** отвара форму за уплату рате аранжмана. Избором дестинације из прве падајуће листе, селектују се и у другој листи приказују сви продати аранжмани за ту дестинацију.

За сваку дестинацију у првој листи види се назив дестинације и држава, док се за сваки продати аранжман види име и презиме клијента и број уговора који идентификује купљени аранжман. Избором у доњој падајућој листи читају се све уплате за изабрани аранжман и приказују у **listView** контроли. Истовремено се испод ње приказује укупна цена аранжмана и износ преосталог дуговања.

У контроли за приказ датума, избором аранжмана, приказује се временски интервал трајања аранжмана.

**Plaćanja**

Destinacija	Granada, Španija																																																	
Aranžman	(ugovor br.20) Ivana David																																																	
Trajanje aranžmana	<p style="text-align: center;">jul 2021.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>pon</td><td>uto</td><td>sre</td><td>čet</td><td>pet</td><td>sub</td><td>ned</td></tr> <tr><td>28</td><td>29</td><td>30</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr> <tr><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td></tr> <tr><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr> <tr><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td></tr> <tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> <p style="text-align: center;">Today: 10.7.2021.</p>	pon	uto	sre	čet	pet	sub	ned	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8
pon	uto	sre	čet	pet	sub	ned																																												
28	29	30	1	2	3	4																																												
5	6	7	8	9	10	11																																												
12	13	14	15	16	17	18																																												
19	20	21	22	23	24	25																																												
26	27	28	29	30	31	1																																												
2	3	4	5	6	7	8																																												
Rata	Uplaćeno	Datum uplate																																																
1	100,00	08.03.2021																																																
2	80,00	25.04.2021																																																
3	80,00	05.06.2021																																																
4	200,00	09.07.2021																																																

**Ukupna cena: 788,00**

**Iznos duga: 328,00**

**Iznos za uplatu:**

**Uplati**

Уносом вредности за уплату и кликом на дугме **Уплати**, врши се уплата наредне рате, упачена рата се приказује у листи, а износ дуговања прерачунава и приказује се нова вредност дуга.

**Plaćanja**

Destinacija	Granada, Španija																																																	
Aranžman	(ugovor br.20) Ivana David																																																	
Trajanje aranžmana	<p style="text-align: center;">jul 2021.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr><td>pon</td><td>uto</td><td>sre</td><td>čet</td><td>pet</td><td>sub</td><td>ned</td></tr> <tr><td>28</td><td>29</td><td>30</td><td>1</td><td>2</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr> <tr><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td></tr> <tr><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr> <tr><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td></tr> <tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> </table> <p style="text-align: center;">Today: 10.7.2021.</p>	pon	uto	sre	čet	pet	sub	ned	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8
pon	uto	sre	čet	pet	sub	ned																																												
28	29	30	1	2	3	4																																												
5	6	7	8	9	10	11																																												
12	13	14	15	16	17	18																																												
19	20	21	22	23	24	25																																												
26	27	28	29	30	31	1																																												
2	3	4	5	6	7	8																																												
Rata	Uplaćeno	Datum uplate																																																
1	100,00	08.03.2021																																																
2	80,00	25.04.2021																																																
3	80,00	05.06.2021																																																
4	200,00	09.07.2021																																																
5	200,00	10.07.2021																																																

**Ukupna cena: 788,00**

**Iznos duga: 128,00**

**Iznos za uplatu:**  200

**Uplati**

Друга команда и дугме са почетне форме отварају преглед дуговања за путовања у предстојећем периоду. Треба да се прикажу сви аранжмани који ће започети у наредних 15 дана и за сваки се поред броја уговора, имена клијента, дестинације, датума поласка и укупне цене, приказује и висина уплаћеног износа, као и преостало дуговање. Поља која показују да клијент има дуг обојити црвено, док за клијенте који су претплатни, поље дуга обојити зелено.

**Spisak putnika u narednom periodu...**

Spisak putnika čiji aranžman počinje u narednih 15 dana i njihova dugovanja:

UgovorID	Ime	Prezime	Destinacija	Polazak	Cena	Placeno	Dugovanje
91	Ivana	Aleksić	Zakintos	23.7.20...	4703	4800	-97
289	Ana	Bogavac	Rim	17.7.20...	4031	3230	801
38	Altan	David	Kapado...	18.7.20...	3120	2950	170
8	Emina	Antić	Granada	20.7.20...	2725	2800	-75
395	Nemanja	Bošković	London	15.7.20...	2562	2562	0
589	Bobana	Andrić	Madrid	23.7.20...	3035	2450	585
455	Viktor	Erceg	Zakintos	20.7.20...	2768	2400	368
435	Aleksan...	Borovac	Zakintos	17.7.20...	3361	2000	1361
482	Milutin	Haidomen	Madrid	23.7.20...	2204	1050	1154

(polasci od 10.07.2021 do 25.07.2021)

Треће (инфо) дугме почетне форме отвара кратко корисничко упутство, а четврто дугме затвара целу апликацију.

Осмислити проширење базе којим се омогућава чување података о смештају у хотелу. За хотелске собе треба памтити број собе, спратност, број кревета, тип собе и опис. Типови соба се чувају као засебан ентитет у бази. Свака соба може бити једног типа (студио, апартман, дуплекс...).

Када су плаћања у питању, потребно је омогућити да се за сваку уплату, у форми додатног описа, региструје начин реализације уплате (кеш, картица, чекови...)

Креирати дијаграм и бекап стања проширене базе.

РАЦИЈА ВЕРЗИЈА

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Туристички обиласци**

Туристичка агенција располаже базом у којој се региструју подаци о свим склопљеним уговорима продаје туристичких аранжмана. Агенција жали да својим клијентима подуди додатне обиласке током боравка на одређеној туристичкој дестинацији. Из сваког одмаралишта (дестинације), постоји понуда одласка на више различитих додатних обилазака (експулзија). Такође, на обиласак неког локалитета могуће је понудити и са више оближњих дестинација. На сваку од експулзија може се ићи више пута током недеље и памте се дани у којима се организује конкретна понуда неке експулзије. Апликацији је потребно додати функционалности за унос и корекцију података о експулзијама – цене и дана извођења

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

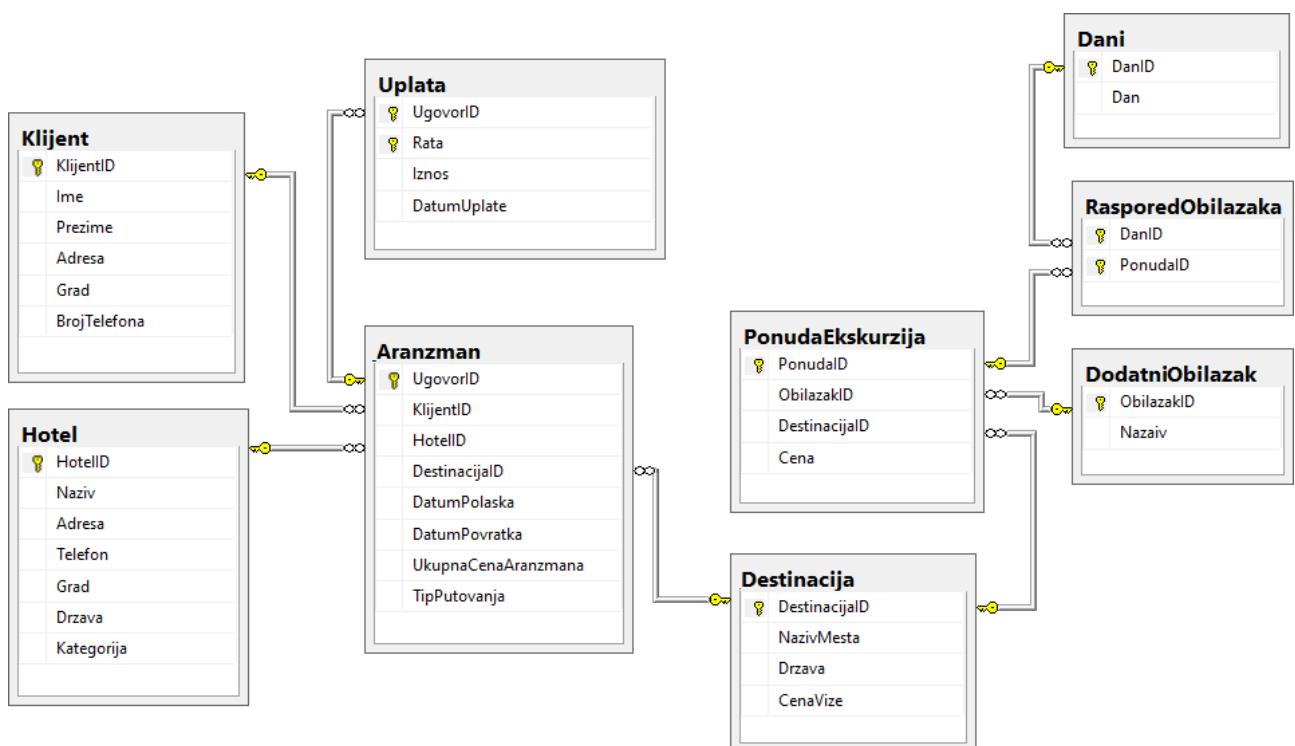
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A211**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A212**.

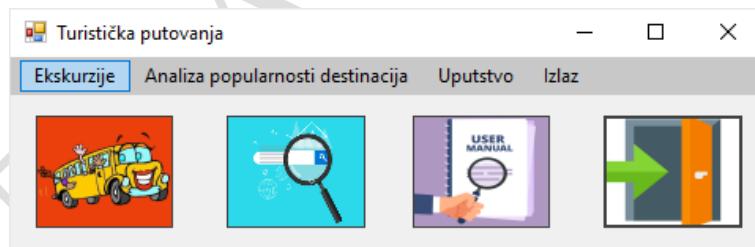
Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:

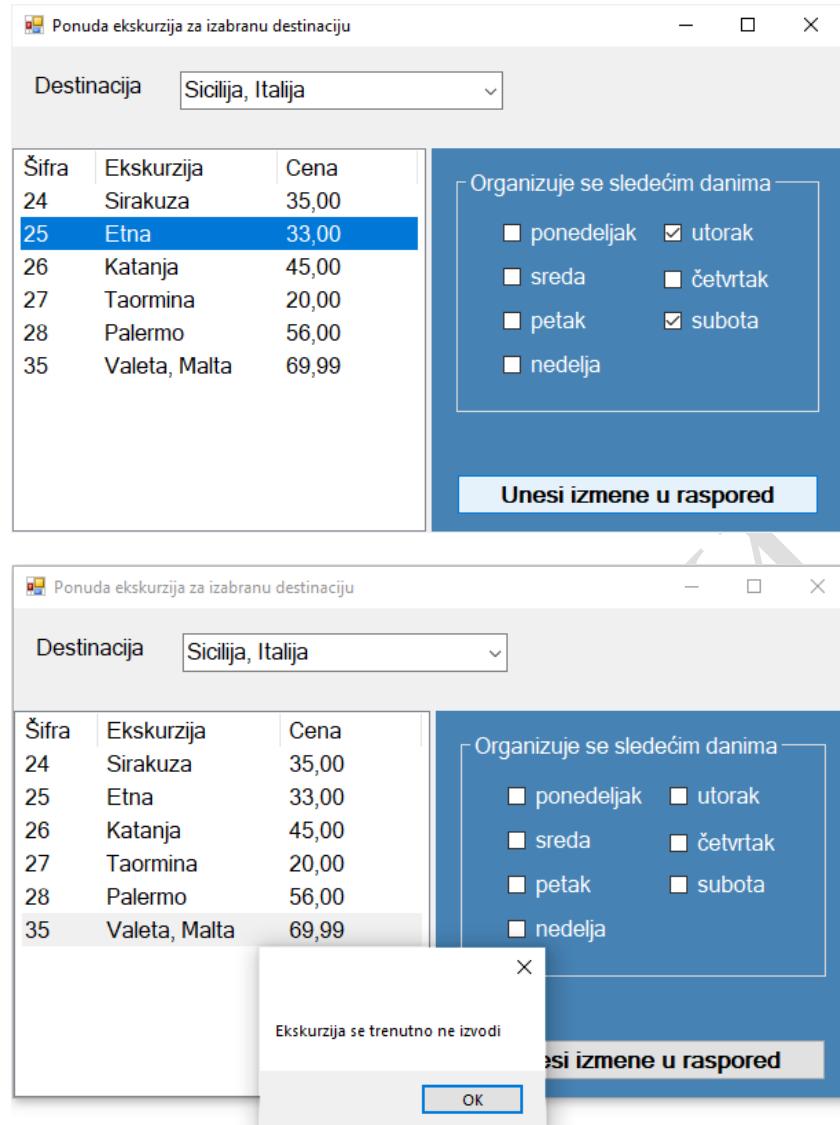


Развојним планом апликације предвиђено је да се уради следеће:

Креирати почетну форму са које се било кликом на одговарајуће дугме или кликом на ставку менија приступа појединачним функционалностима апликације.



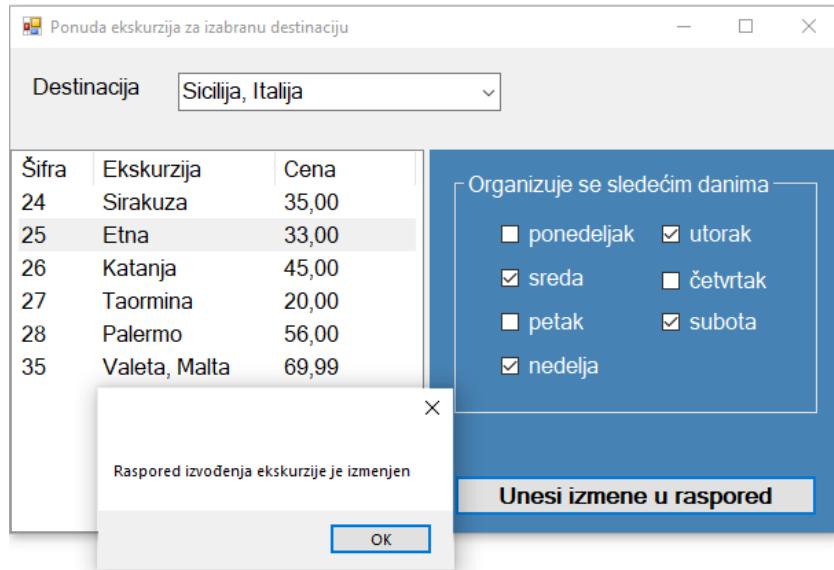
Прво дугме **Екскурзије** отвара форму за преглед понуда екскурзија за сваку од дестинација у понуди.



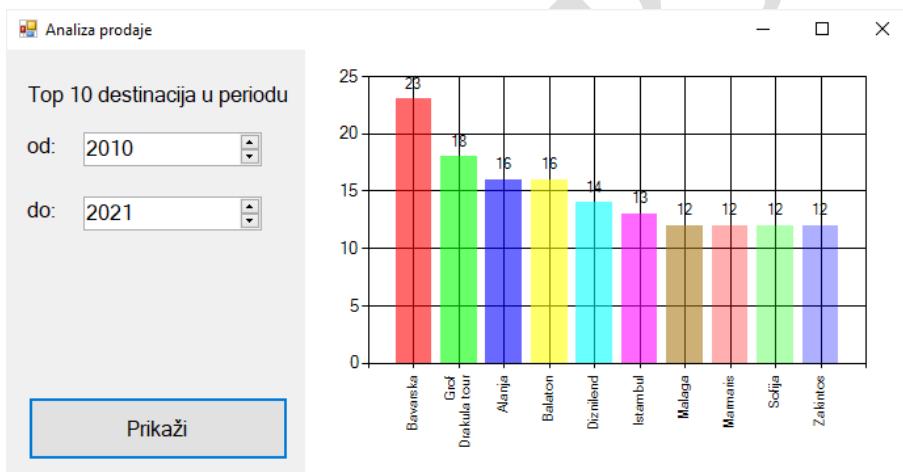
Избором дестинације из падајуће листе, у *ListView* контроли се приказују све понуде акскурзија са те десницације. За сваку понуду приказује се ширфа понуде, назив екскурзије и цена. Кликом на ставку са подацима о екскурзији у десном делу форме са чекирају дани у недељи када се екскурзија изводи за туристе на изабраној дестинацији.

Уколико су сви дани дечекирани, треба избацити поруку „**Екскурзија се тренутно не изводи**“

Пошто често долази до промена распореда екскурзија, потребно је омогућити да се чекирањем и дечекирањем обележи у којим данима екскурзија иде и да се унете промене упису у базу и шаље се обавештење о насталој промени.



Друга команда и дугме са почетне форме отварају графички преглед 10 најпродаванијих дестинација у задатом опсегу година. Имајући у виду да је агенција почела са радом пре 30 година, обезбедити да корисник апликације не може да унесе године ван дозвољеног опсега.



Треће (инфо) дугме почетне форме отвара кратко корисничко упутство, а четврто дугме затвара целу апликацију.

Проширити базу тако да се за сваку понуду екскурзије памти и време када се креће на екскурзију и очекивано време повратка.

Такође, аранжмани на некој дестинацији се продају у сарадњи са домаћим и страним операторима. На једној дестинацији, агенција сарађује само са једним тур-оператором, док један тур-оператор може сарађивати са већим бројем агенција.

Креирати дијаграм и бекап стања проширене базе.

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Голф куп**

За потребе организације и регистраовања резултата голф мечева током надметања на међународном турниру у овом спорту, креирана је база података и у току је развој апликације за приступ и манипулацију базом. Централни део посла односи се на развој логике за заказивање сусрета, тј који играчи, у које време и на ком терену играју меч. Други део посла односи се на логику која би се користила за преглед мечева који су заказани на неком терену.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

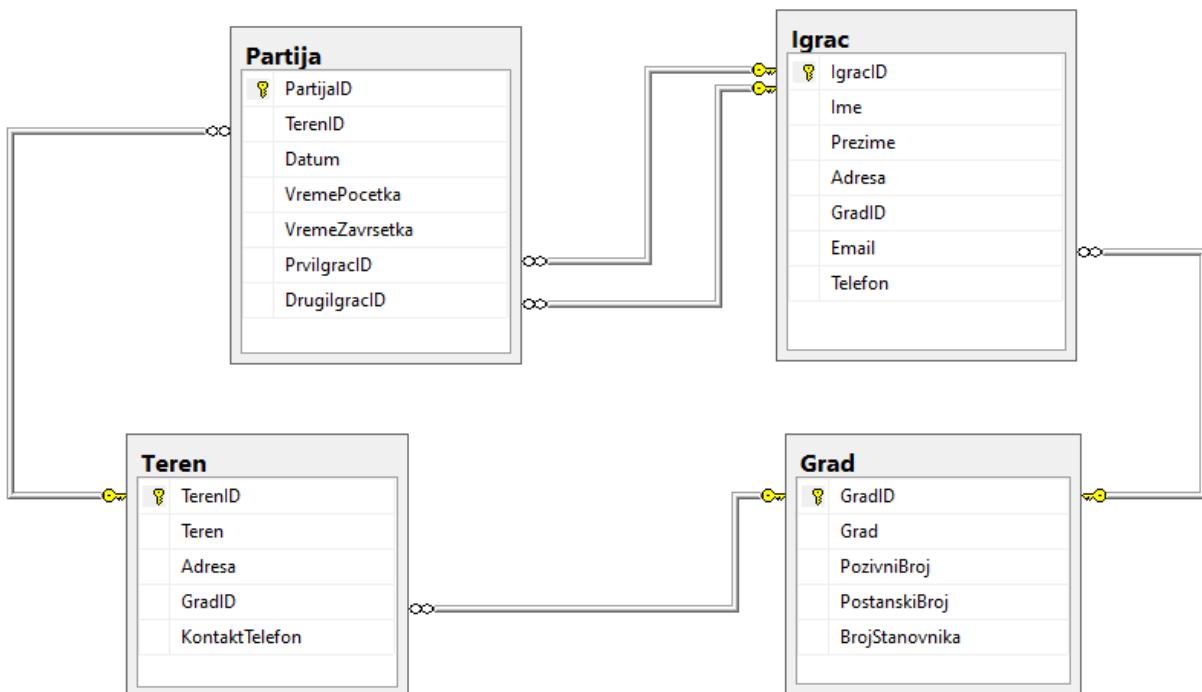
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A221**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A222**.

Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



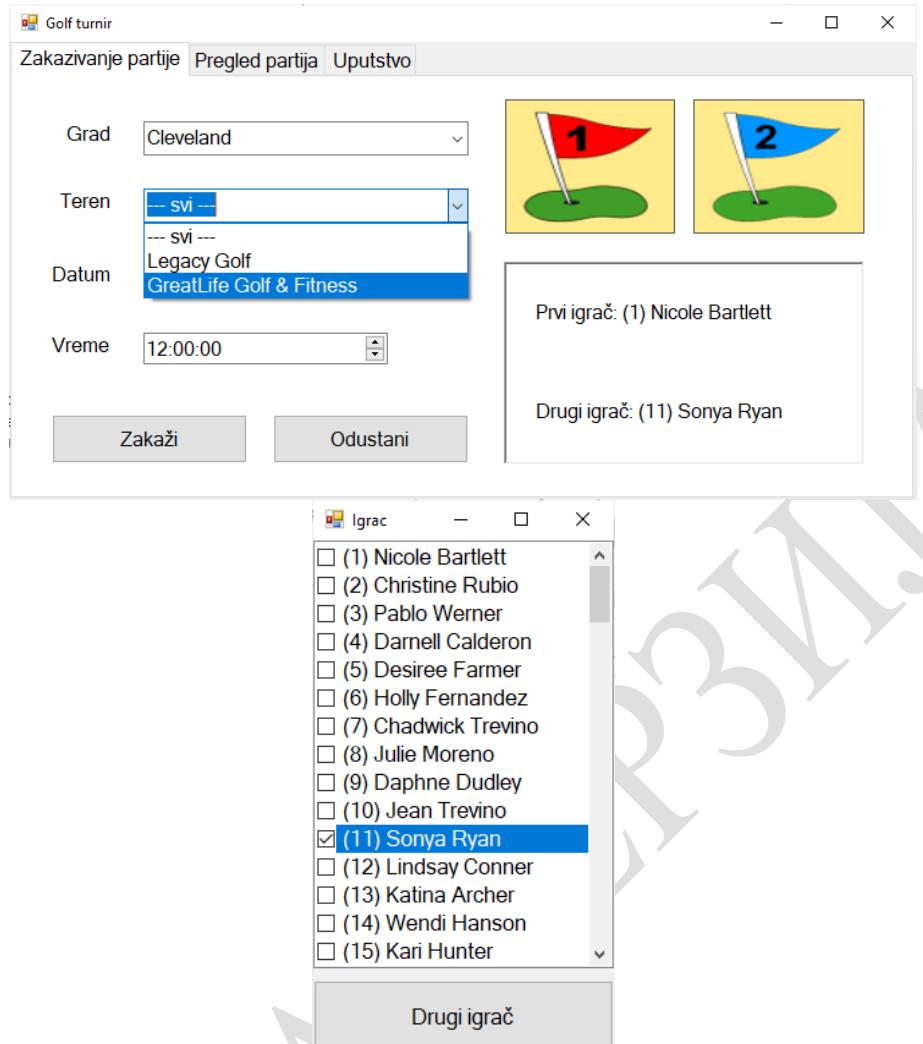
Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која има три странице (таба). На првој се врши заказивање партија тако што се прво изабере град у коме се игра. Ставка **-сви-** ресетује стање у падајућим листама. Избором града, у другој падајућој листи се приказују терени који се налазе у изабраном граду. У контролама за избор датума и времена одређује се када се меч игра.

Обезбедити да се меч може заказати најкасније 24 часа пре почетка меча.

Дугме са slikом заставице оуначене бројем 1, отвара помоћну форму за одабир првог играча. Помоћна форма приказује све играче, а играч се бира чекирањем у листи и кликом на дугме **Први играч**.

Клик на друго дугме, отвара форму за избор другог играча. Обезбедити да не може бити изабран исти играч на месту оба учесника сусрета.



Уколико су сви параметри партије правилно одабрани, кликом на **Закажи** нови сусрет се уписује у базу (код парија које су заказане, а нису још одигране, поље у бази које чува време завршетка је **NULL**). Приказати поруку о успешности уписа.

Дугме **Одустани** затвара апликацију.

Преглед одиграних и заказаних партија се врши на другом табу. Одговарајућим дугметом за избор, бира се да ли треба приказати завршене или заказане, а још увек не одигране партије. Дугме **Прикажи** у горњој контроли (**dataGridView**) приказује терене и број партија на сваком терену. Кликом на неки од редова у горњој контроли, у дољој се приказују сви парови и датуми одржавања мечева.

Golf turnir

Zakazivanje partije Pregled partija Uputstvo

Prikaži  
 zakazane partije  
 odigrane partije

TerenID	Teren	Grad	Broj
5	Boulder Creek Golf...	Chicago	2
33	GreatLife Golf & Fi...	Cleveland	5
26	Maxwell Municipal ...	Indianapolis	7
9	Ruggles Golf Club	Jacksonville	3
4	Cedar River Countr...	Lubbock	5
10	Silver Lake Countr...	Nashville	1
7	Grays Harbor Golf ...	Newark	6
14	Abita Springs Golf ...	Tulsa	5

Igrači	Datum	Početak
Holly Fernandez vs Abigail Kra...	02 Oct 202	08:19:17
Jarrod George vs Claudia Fiqu...	21 Sep 2...	00:44:43
Roger Duffy vs Angela Daniel	26 Oct 202	08:23:27
Devon Henry vs Brian Walker	23 Sep 2...	02:08:59
Nicole Bartlett vs Sonya Ryan	31 Jul 202	12:00:00
*		

Prikaži
Izadi

На трећој страници (табу) налази се кратка пратећа техничка документација са описом структуре и функционалности апликације.

У међувремену је дошло до проширења захтева, тако да је и базу потребно проширити тако да се чувају подаци о главном судији сваког сусрета. За сваког судију се памти име, презуме, датум стицања лиценце, град из кога долази. Свака партија има само једног главног судију.

Такође, за сваку партију треба додатно памтити и детаље о временским условима под којима је партија одиграна.

Креирати дијаграм и бекап стања проширене базе.

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Позоришне представе**

Позоришна кућа је исказала потребу за апликацијом којом би се регистровале резервације карата за најављене позоришне представе. Кориснику је потребно омогућити преглед репертоара у неком изабраном временском опсегу. При томе, треба омогућити преглед свих предстојећих представа или само представе изабраног позоришног комада.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

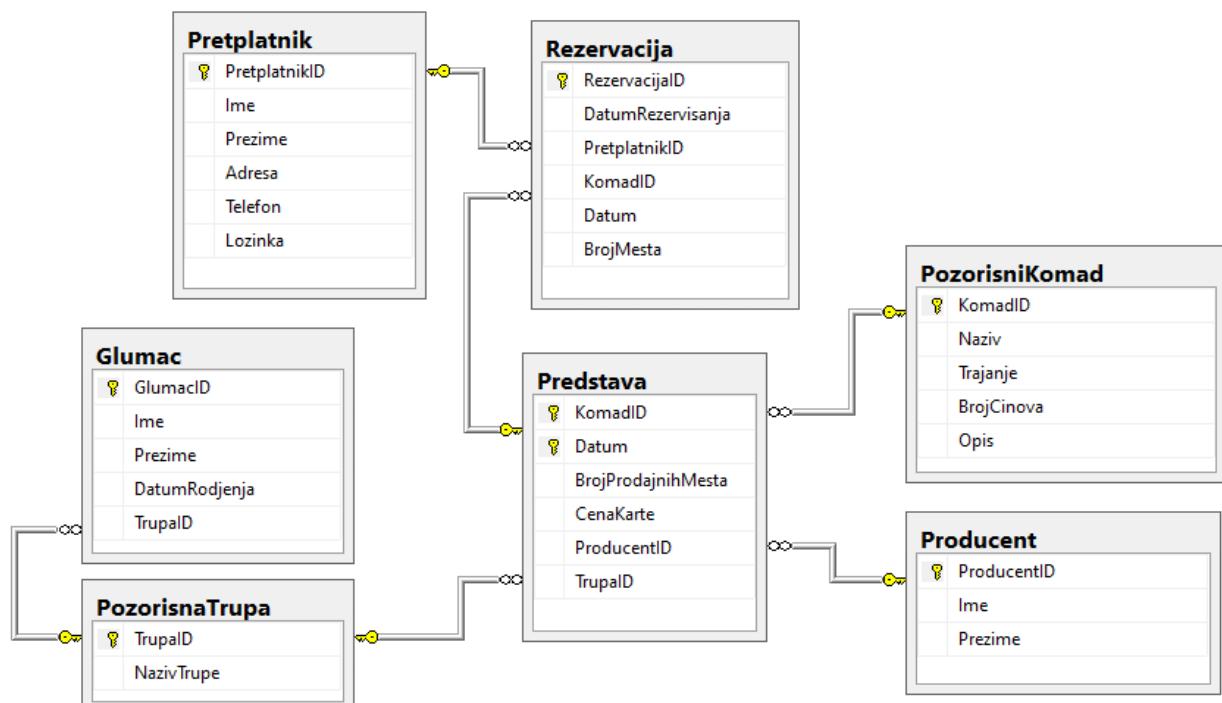
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A231**.

Резервна копија почетног стања базе дате је као **Прилог\_4ЕИТ\_A232**.

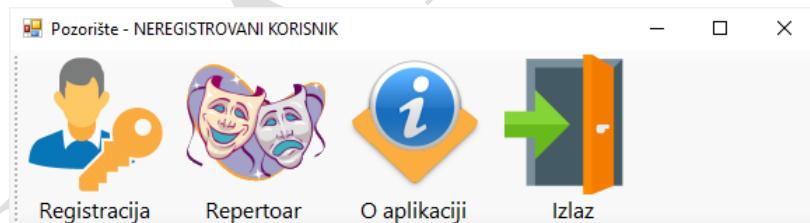
Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:

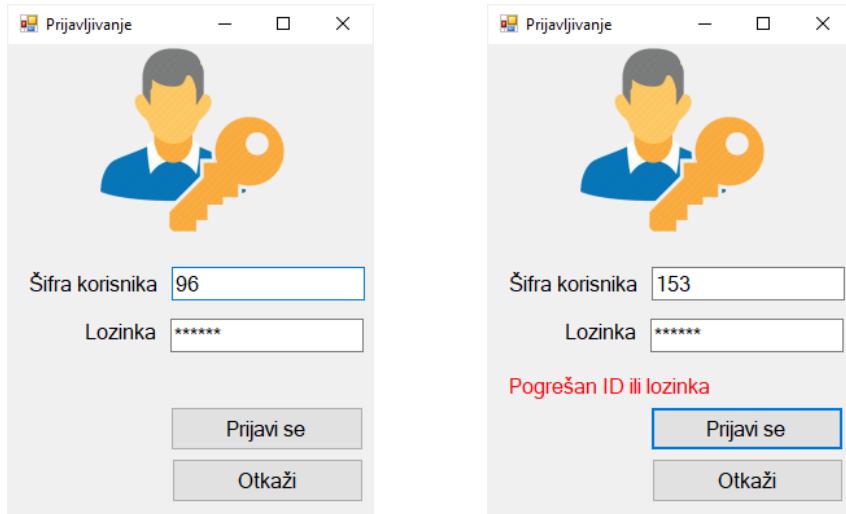


Развојним планом апликације предвиђено је да се уради следеће:

Креирати почетну форму са које се кликом на дугме на линији са алатима отвара одговарајућа форма. Прво дугме (**Регистрација**) отвара форму за логовање корисника.



Уноси се шифра корисника и лозинка из базе и ако је корисник нађен, стартни натпис у насловној линији се мења и уместо „**НЕРЕГИСТРОВАНИ КОРИСНИК**”, приказује се текст „**РЕГИСТРОВАНИ КОРИСНИК**“. У супротном, само се приказује порука „**Погрешан ИД или лозинка**“.



Дугме (**Репертоар**) отвара форму за преглед представа које ће бити одржане у наступајућем периоду и омогућава резервацију карата. Ова форма се може отворити и ако корисник није логован:

- Ако је логован, приказују се подаци о логованом кориснику.
- Ако није логован, могућа је претрага, али не и резервација. Да би се извршила резервација мора прво да се логује, што може и са ове форме. (Ако се кликне команда за регистрацију у стању када је неко већ регистрован, врши се аутоматско одјављивање пријављеног корисника).

Ако корисник кликне дугме **На репертоару**, а при томе није одређен датум, нити изабран комад, приказују се све заказане представе почев од данашњег дана, па на даље, сортиране по датумима.

Кликом у текст поље за унос датума, отвара се контрола **MonthCalendar** на којој је могуће изабрати опсег датума за који се врши претрага представа. Када се после избора, кликне било где на форму (ван календара) опсег се очитава и уписује у предвиђено текст поље.

Избор датума сужава претрагу само на позоришне комаде чије је извођење заказано у задатом периоду и њима се попуњава падајућа листа **Комад**. Избором комада из падајуће листе, у контроли **dataGridView** приказују се све заказане представе изабраног позоришног комада у означеном периоду.

Repertoar - REGISTROVANI KORISNIK

	Datum	Komad	
	od 14.07.2021 do 12.08.2021	Edmun Kin	<a href="#">Na repertoaru &gt;&gt;</a>
*	5	Edmun Kin	20.7.2021 Atelje 212
*	5	Edmun Kin	11.8.2021 Narodno pozorište Niš
*	5	Edmun Kin	12.8.2021 Beogradsko dramsko p...

Preplatnik:  
Una Filipović  
(id: 96)

Broj mesta

Cena karte

Za uplatu

Prijavite se

Rezerviši

Izadi

Дугме **Резервиши**, врши резервацију задатог броја карата за селектовану представу. Корисник се прво упозорава да ће извршити резервацију и предочава му се укупна цена.

Repertoar - REGISTROVANI KORISNIK

	Datum	Komad	
	od 14.07.2021 do 12.08.2021	Edmun Kin	<a href="#">Na repertoaru &gt;&gt;</a>
*	5	Edmun Kin	20.7.2021 Atelje 212
*	5	Edmun Kin	11.8.2021 Narodno pozorište Niš
*	5	Edmun Kin	12.8.2021 Beogradsko dramsko p...

Preplatnik:  
Una Filipović  
(id: 96)

Broj mesta

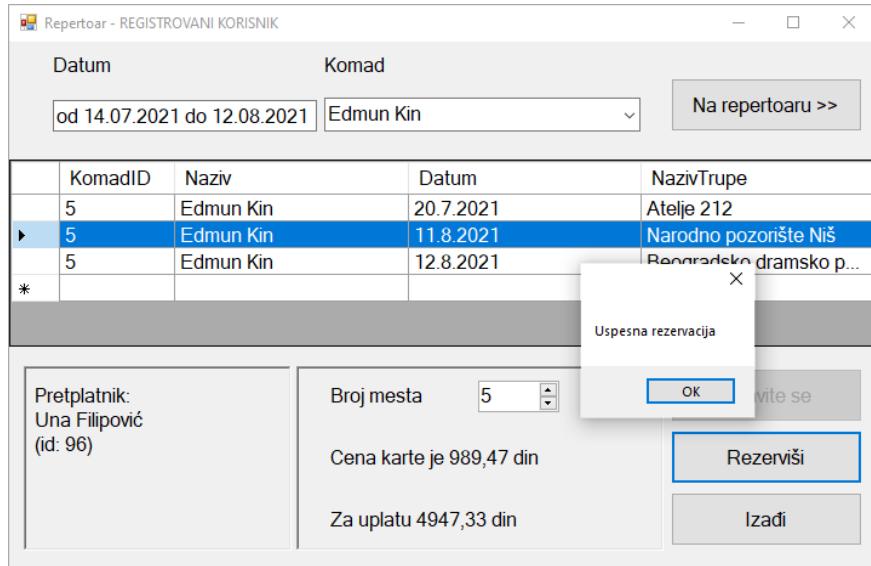
Cena karte

Za uplatu

PAZNJA

Ovim se slažete sa kupovinom:  
broj karata 5  
iznos za plaćanje: 4947,332

Ако одговори са **YES** у дијалог прозору (**MessageBox**) који се отвара, врши се упис резервације и приказује цена на форми. Ако је одговор на дијалог **NO**, не уписује се резервација у базу, већ се може одустати кликом на **Изађи** или се могу променити параметри за резервацију.



Треће дугма на почетној форми отвара кратку пратећу техничку документацију са описом структуре и функционалности апликације. Четврто дугме затвара апликацију.

У међувремену је дошло до проширења захтева, тако да је и базу потребно проширити тако да се чувају подаци о аутору позоришног комада: име, презиме, датум рођења, као и година када је дело по коме је постављен позоришни комад написано. Такође је потребно омогућити регистровање поена који се кориснику додељују уз сваку обављену куповину на основу којих је касније могуће остварити одређене повластице при наредним куповинама.

Креирати дијаграм и бекап стања проширене базе.

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Поликлиника**

Већи број поликлиника функционише на различитим локацијама и у различитим градовима. Лекари различитих специјалности могу бити ангажовани на једној или више поликлинике истовремено. Ради лакшег заказивања прегледа, креирана је база где се памти који је лекар ангажован на којој клиници. База памти и историју ангажовања лекара, тј. поред податка о томе где тренутно ради, могуће је добити и информацију на којим клиникама је претходно радио. Такође, имати у виду да је могуће лекара који је некад радио, па раскинуо уговор, поново ангажовати на истом месту.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

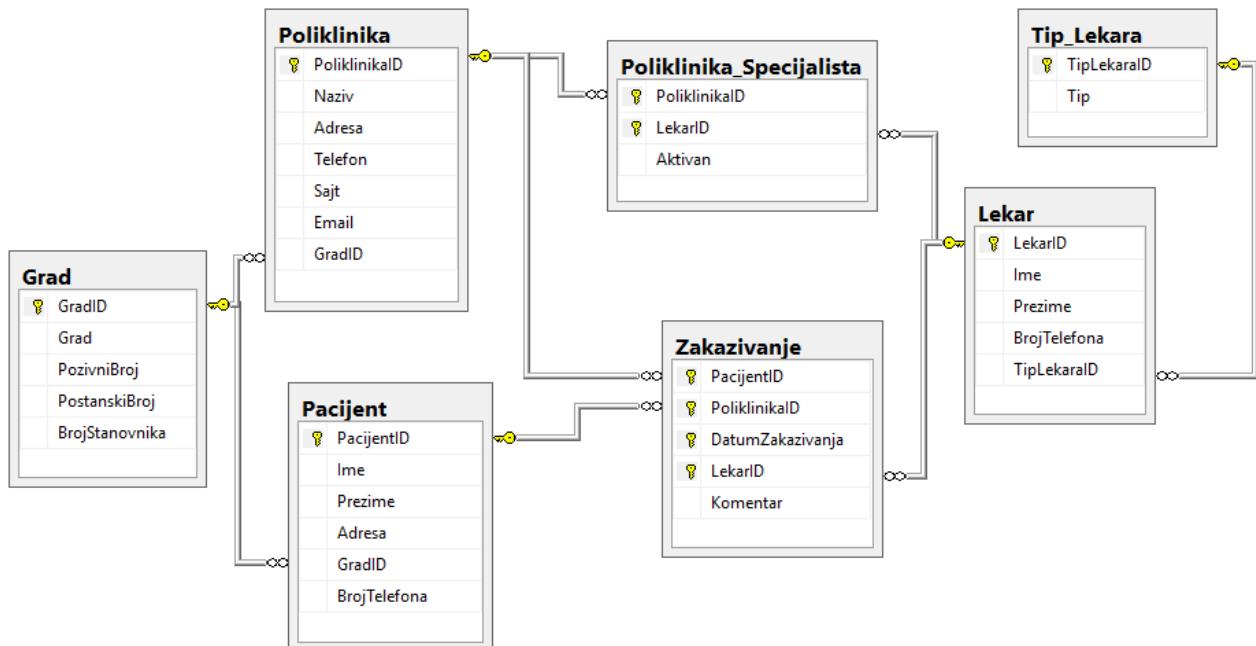
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A241**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A242**.

Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која има три странице (таба). На првој се врши распоређивање лекара-специјалиста у жељене клинике тако што се прво изабере специјализација, а затим и лекар кога треба ангажовати или разрешити ангажовања у једној или више клиника.

На почетку, прва падајућа листа попуњена је свим специјализацијама које постоје у бази, док се у другој налази списак свих лекара. Избором специјализације, у другој падајућој листи се приказују само лекари специјалисти изабране области чиме се олакшава претрага лекара.

Grad	Poliklinika	Status
Valjevo	Poliklinika Bebafa...	<input type="checkbox"/>
Novi Sad	Poliklinika prezident	<input type="checkbox"/>
Beograd	Ultrazvuk	<input type="checkbox"/>
Prokuplje	Poliklinika Nada D...	<input type="checkbox"/>
Paraćin	Poliklinika Djukić	<input type="checkbox"/>
Prokuplje	Poliklinika Naissa	<input type="checkbox"/>
Čačak	Bel medic opšta b...	<input type="checkbox"/>
Vrbas	Poliklinika Bocokic	<input type="checkbox"/>
Negotin	Poliklinika alfa me...	<input type="checkbox"/>
Kraljevo	Opšta bolnica imp...	<input type="checkbox"/>
Kraljevo	Poliklinika dr Marić	<input type="checkbox"/>
Apatin	Dom zdravlja dr Ri...	<input type="checkbox"/>
Vranje	Poliklinika Human	<input type="checkbox"/>
Apatin	Poliklinika ginako	<input type="checkbox"/>
Negotin	Poliklinika ars me...	<input type="checkbox"/>
Vlasotince	Ordinacija Vuk Pri...	<input type="checkbox"/>
Smederevo	Poliklinika Corona	<input type="checkbox"/>
Negotin	Poliklinika Dr Nikolić	<input type="checkbox"/>
Užice	Poliklinika OMT	<input type="checkbox"/>

Тек када се изаберу и специјализација и лекар, у контроли за приказ података (**dataGridView**) приказује се списак поликлиника на којима је ангажован изабрани лекар (приказан је списак свих поликлиника, а оне у којима изабрани лекар ради, штиклиране су у колони **Статус**).

Grad	Poliklinika	Status
Niš	Klinika Una	<input checked="" type="checkbox"/>
Užice	Mirabilis	<input type="checkbox"/>
Sombor	Poliklinika i bolnic...	<input type="checkbox"/>
Pančevo	Milmedic poliklinika	<input type="checkbox"/>
Paraćin	Poliklinika panacea	<input type="checkbox"/>
Apatin	Alba poliklinika	<input type="checkbox"/>
Zaječar	Poliklinika intertim	<input type="checkbox"/>
Negotin	Magna Plus	<input type="checkbox"/>
Vranje	Dr Djoric	<input type="checkbox"/>
Beograd	Vivalab	<input type="checkbox"/>
Vlasotince	Poliklinika argirović	<input type="checkbox"/>
Subotica	Demetra	<input type="checkbox"/>
Novi Pazar	Life Impuls	<input type="checkbox"/>
Niš	Medikom poliklinika	<input checked="" type="checkbox"/>
Vršac	Poliklinika anlave	<input type="checkbox"/>
Valjevo	Sava surgery	<input type="checkbox"/>
Leskovac	Poliklinika antame...	<input checked="" type="checkbox"/>
Kraljevo	Poliklinika Dobrić	<input type="checkbox"/>
Čačak	Dom zdravlja vizim	<input type="checkbox"/>

Омогућити доделу ангажовања лекару тако што се штиклира поље статуса у одговарајућем реду. Разрешавање дужности на некој клиници врши се дечекирањем поља статуса. Лекара је могуће истовремено разрешити дужности и ангажовати на једној или више клиника.

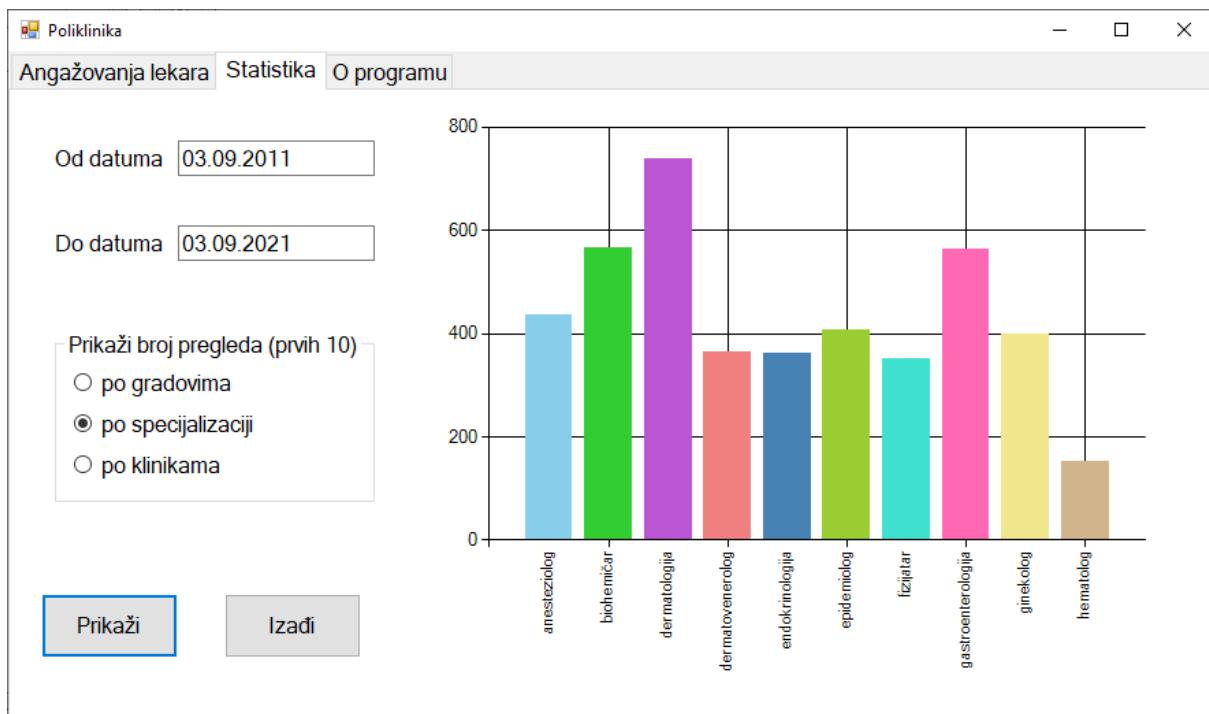
- Приликом разрешавања дужности, не брише се ставка из базе (у циљу чувања историје ангажовања), већ се само ажурира статус са „активан“ на „није активан“.
- Уколико се лекар први пут ангажује на клиници, уписује се ставка у табелу. Уколико је раније (у једном или више наврата) био ангажован па премештан, потребно је само ажурирати статус са „није активан“ на „активан“.

Grad	Poliklinika	Status
Niš	Klinika Una	<input checked="" type="checkbox"/>
Užice	Mirabilis	<input type="checkbox"/>
Sombor	Poliklinika i bolnic...	<input type="checkbox"/>
Pančevo	Milmedic poliklinika	<input type="checkbox"/>
Paraćin	Poliklinika panacea	<input type="checkbox"/>
Apatin	Alba poliklinika	<input type="checkbox"/>
Zaječar	Poliklinika intertim	<input checked="" type="checkbox"/>
Negotin	Magna Plus	<input checked="" type="checkbox"/>
Vranje	Dr Djoric	<input type="checkbox"/>
Beograd	Vivalab	<input type="checkbox"/>
Vlasotince	Poliklinika argirović	<input type="checkbox"/>
Subotica	Demetra	<input type="checkbox"/>
Novi Pazar	Life Impuls	<input type="checkbox"/>
Niš	Medikom poliklinika	<input checked="" type="checkbox"/>
Vršac	Poliklinika anlave	<input type="checkbox"/>
Valjevo	Sava surgery	<input type="checkbox"/>
Leskovac	Poliklinika antame...	<input type="checkbox"/>
Kraljevo	Poliklinika Dobrić	<input type="checkbox"/>
Čačak	Dom zdravlja vizim	<input type="checkbox"/>

Дугме **Изађи** затвара апликацију.

На картици **Статистика** треба графички представити број прегледа у дефинисаном временском периоду. Уколико временски интервал није унет, кликом на дугме **Прикажи** треба приказати статистику за последњих 10 година.

Радио дугмићима се одређује по ком критеријуму ће се извршити пребројавање прегледа. На графику треба приказати првих 10 највећих вредности и то сортираних по алфабетском редоследу.



На трећој страници (табу) налази се кратка пратећа техничка документација са описом структуре и функционалности апликације.

Исказана је потреба за проширењем базе због обавезе да лекари похађају семинаре стручног усавршавања и потребе да се о томе води евиденција. Сваки програм стручног усавршавања се карактерише својим називом и описом. Потребно је регистровати које је програме усавршавања завршио сваки од лекара, као и датум када је завршио обуку.

За сваког лекара додати податак имејл са провером унесености правилне адресе.

Креирати дијаграм и бекап стања проширене базе.

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–A25**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Поликлиника**

Поликлиника има своје зграде (пословнице) на више различитих локација. Пословање је организовано тако да на поликлиници ради већи број лекара, од којих неки, по потреби могу радити и на већем броју различитих локација. Да не би дошло до грешке у заказивању, креирана је база која памти све посете пациентата: који доктор, на којој локацији и код датума је примио конкретног пацијента. За сличај да је доктор изненада недоступан, потребно је додати функционалност презаказивања прегледа за други термин. Уколико пациенту не одговара презаказивање, потребно реализовати отказивање прегледа. Једном заказане прегледе не треба брисати, већ их треба чувати у бази, са тим што се памти статус (обављен, отказан или null ако је заказан, али још није стигао термин прегледа).

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

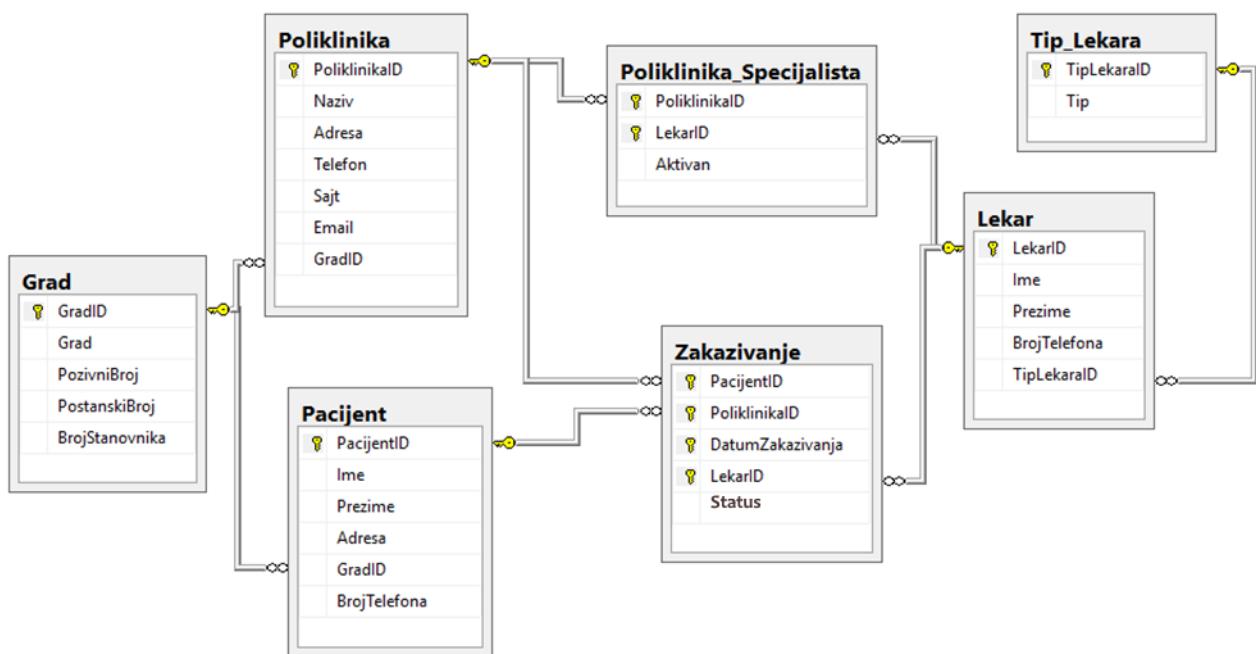
Прилог преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A251**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A252**.

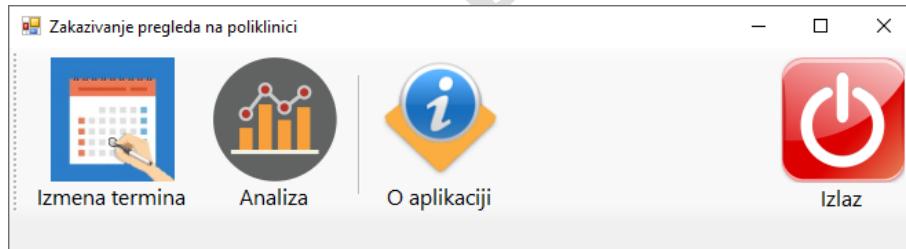
Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати стартну форму која има мени са четири дугмента. Прво дугме почетне форме отвара форму Пацијенти на којој се врши промена термина и отказивање заказаних термина.



У падајућу листу попуњавају се шифре (ЛБО) пацијената. Шифра се може изабрати из листе или укуцати директно у поље падајуће листе. При уносу укуцањем, уколико унета шифра не постоји, приказује се порука „**Непостојећи ЛБО**“

Уколико је унета постојећа шифра или изабрана из падајуће листе, кликом на дугме са иконом лупе, приказују се подаци о пацијенту у текст пољу, а у десном делу форме, на контоли **listview** приказују се подаци о свим заказивањима изабраног пацијента сортирани растуће по датумима (на којој клиници, код ког доктор, ког датума и статус заказаног прегледа: обавељен, отказан или без статуса, ако још није стигао термин прегледа).

The screenshot shows the 'Pacijenti' application interface. On the left, there is a sidebar with 'Podaci o pacijentu' containing a dropdown menu set to '965' and a search icon. Below it is a box with patient details: LBO: 965, Name: Nataša Apostolović, Address: Radoja Dakića, 94, Valjevo, Phone: 061-365-4808. The main area contains a table titled 'Klinika' listing various clinics with their names, doctors, dates, and status. One appointment for 'Poliklinika uromedica' is selected. At the bottom, there are buttons for 'Promeni datum' (Change date), 'Otkazivanje' (Cancellation), and 'Izlaz' (Exit).

Klinika	Lekar	Datum	Status
Poliklinika argirović	Zorana Avramović	25.08.2018	pregled ob.
Sava surgery	Mihajlo Filipović	22.06.2020	pregled ob.
Poliklinika intertim	Jelena Bebić	15.07.2020	otkazano
Poliklinika Human	Jerina Branković	07.11.2020	pregled ob.
Poliklinika Corona	Nikša Dimitrijević	15.02.2021	pregled ob.
Poliklinika ars medica	Kristina Avramović	21.09.2021	otkazan
<b>Poliklinika uromedica</b>	<b>Tamara Beloica</b>	<b>24.10.2021</b>	
Poliklinika Nada Diva	Aleksa Drenovac	14.11.2021	
Poliklinika Selecta	Brankica Aleksić	28.12.2021	

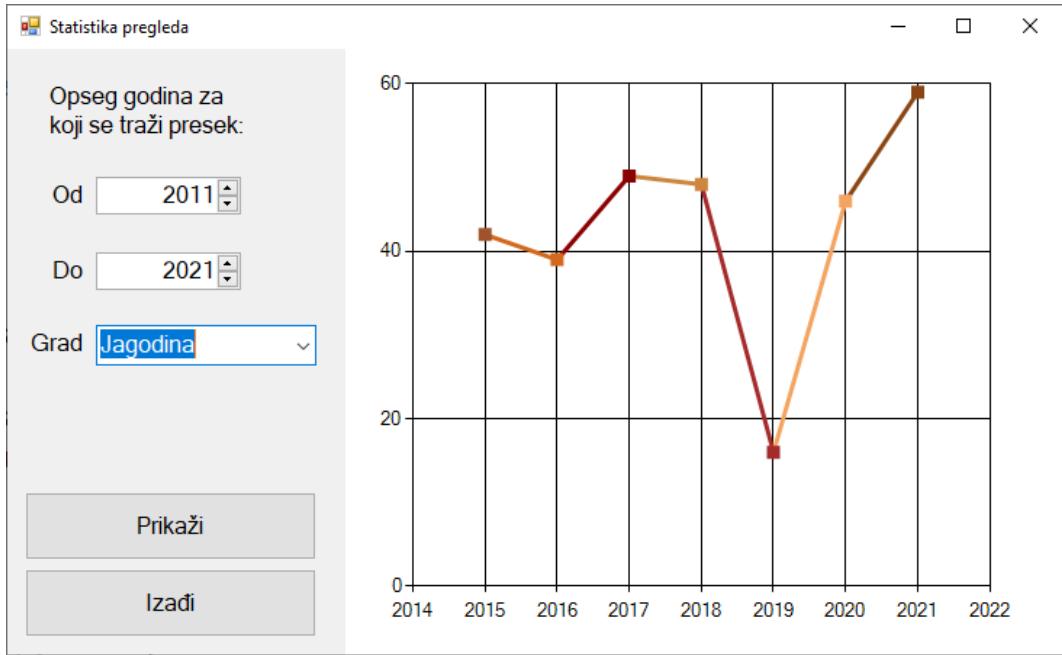
Да би се променио датум или статус, преглед се мора селектовати у листи. Кликом на дугме **Отказивање** уписује се статус „отказан“ и освежава се приказ у листи. Избором новог датума и кликом на дугме **Промени датум**, мења се датум прегледа и освежава листа. Водити рачуна да се промена датума и отказивање могу радити само за прегледе који тек предстоје.

This screenshot shows the same application interface as the previous one, but with a different date selected in the 'Promena datuma' field: 'ponedeljak, 17. januar 2022.' The rest of the interface and data are identical to the first screenshot.

Klinika	Lekar	Datum	Status
Poliklinika argirović	Zorana Avramović	25.08.2018	pregled ob.
Sava surgery	Mihajlo Filipović	22.06.2020	pregled ob.
Poliklinika intertim	Jelena Bebić	15.07.2020	otkazano
Poliklinika Human	Jerina Branković	07.11.2020	pregled ob.
Poliklinika Corona	Nikša Dimitrijević	15.02.2021	pregled ob.
Poliklinika ars medica	Kristina Avramović	21.09.2021	otkazan
Poliklinika Nada Diva	Aleksa Drenovac	14.11.2021	
Poliklinika Selecta	Brankica Aleksić	28.12.2021	
<b>Poliklinika uromedica</b>	<b>Tamara Beloica</b>	<b>17.01.2022</b>	

Дугме **Излаз** затвара форму, али не и целију апликацију.

Дугме **Анализа** са почетне форме отвара форму за преглед броја заказаних и обављених прегледа за пацијенте из изабраног града. Пресек стања треба начинити за задати опсег година, а ако другачије није задато, за претходних 10 година у односу на текућу. Кликом на **Прикажи** исцртава се графички приказ података, а **Издији** затвара форму **Статистика прегледа**.



На трећој страници (табу) налази се кратка пратећа техничка документација са описом структуре и функционалности апликације. Четврто дугме (*Излаз*) на почетној форми, затвара целу апликацију.

Обзиром да поликлиника креће да отвара своје пословнице и у иностранству, потребно је проширити базу тако да се памти у којој се држави пословница налази. За државу се памти назив и позивни број.

Такође, треба да се омогући регистраовање вакцина које је примио сваки од пацијената. За вакцину се памти назив и произвођач. Имамо у виду да пациент може да прими више вакцина и за сваку примљену вакцину памти се датум када је апликована.

Креирати дијаграм и бекап стања проширене базе.

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Хотелске резервације**

Хотел води евиденцију о начињеним резервацијама и уплатама резервисане услуге. Корисник-уговарач (муштерија) може да резервише једну и више соба жељене структуре и при томе наводи имена осталих корисника аранжмана (гости).

Уговарачу аранжмана је потребно обезбедити могућност плаћања на рате. Број рата није ограничен, али постоји обавеза комплетне исплате аранжмана до уговором предвиђеног датума. Прихватљив је било који начин плаћања појединачних рата (кеш, картице... )

У циљу вођења прецизне евиденције о уплатама и

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

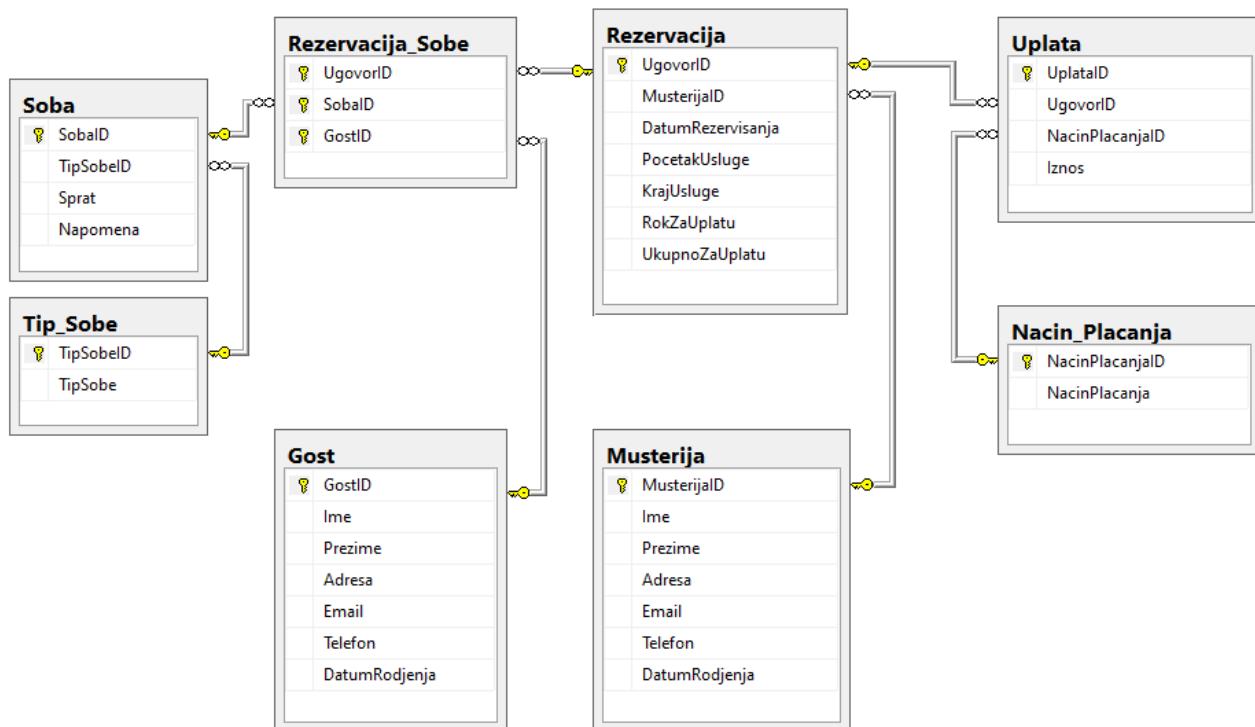
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A261**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A262**.

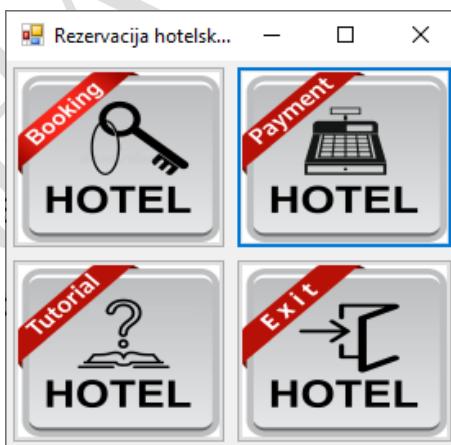
Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати стартну форму која има мени са четири дугмата. Дугме **Payment** почетне форме отвара форму **Плаћања** на којој се врши уплата рате аранжмана.



Избором броја уговора из одговарајуће падајуће листе, у пољу испод листе се приказују подаци о уговорачу и аранжману: име и презиме корисника, почетак и трајање аранжмана, укупна цена, рок за измирење дугова и преостали износ за плаћање.

Кликом на дугме **Уплати** врши се уплата унетог износа. Да би се извршила уплата потребно је поред броја уговора одредити и начин и износ уплате.

По извршеној уплати освежити приказ рата плаћања у листи. У пољу података о уговорачу приказати нови износ дуга.

**Plaćanja**

<b>Broj ugovora</b>	<b>Način plaćanja</b>
277	Izaberite način plaćanja...

**Podaci o ugovaraču**

Ugovor br: 277  
Ugovarač: Alekса Dačić  
Početak usluge: 24.12.2015  
Broj dana: 13  
Ukupan iznos: 4225  
Rok za uplatu: 21.12.2015  
Dug: 2000

**Iznos uplate**

**Zatvori**

Rata	Iznos	Način plaćanja
1	431,00	čekovima građana
2	433,00	čekovima građana
3	455,00	čekovima građana
4	433,00	čekovima građana
5	473,00	gotovinsko

**Plaćanja**

<b>Broj ugovora</b>	<b>Način plaćanja</b>
277	platna kartica

**Podaci o ugovaraču**

Ugovor br: 277  
Ugovarač: Alekса Dačić  
Početak usluge: 24.12.2015  
Broj dana: 13  
Ukupan iznos: 4225  
Rok za uplatu: 21.12.2015  
Dug: 1000

**Iznos uplate**

**Zatvori**

Rata	Iznos	Način plaćanja
2	433,00	čekovima građana
3	455,00	čekovima građana
4	433,00	čekovima građana
5	473,00	gotovinsko
6	1000,00	platna kartica

Дугме **Излаз** затвара форму, али не и целу апликацију.

Дугме **Booking** са почетне форме отвара форму за претрагу слободних капацитета у задатом периоду. Корисник одређује у од кад до кад би желео да резервише собу. Није неопходно да се одреде тачан почетак и крај боравка. Због лакшег проналажења расположивих капацитета, треба омогућити да се уместо конкретног датума може изабрати опсег датума који долазе у обзор као почетак, тј опсег датума за жељени крај боравка. Кликом на **Претрага капацитета** приказују се подаци о расположивим собама (број собе, тип собе и спрат).

Дугме **Затвори** затвара форму **Расположиви капацитети**.

Raspoloživi kapaciteti

Pretraga kapaciteta							Zatvori																																																								
Polazak	<table border="1"> <thead> <tr> <th colspan="7">decembar 2021.</th> </tr> <tr> <th>pon</th><th>uto</th><th>sre</th><th>čet</th><th>pet</th><th>sub</th><th>ned</th> </tr> </thead> <tbody> <tr><td>29</td><td>30</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> <tr><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td></tr> <tr><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td></tr> <tr><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td></tr> <tr><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> </tbody> </table> <p>Today: 8.9.2021.</p>						decembar 2021.							pon	uto	sre	čet	pet	sub	ned	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	
decembar 2021.																																																															
pon	uto	sre	čet	pet	sub	ned																																																									
29	30	1	2	3	4	5																																																									
6	7	8	9	10	11	12																																																									
13	14	15	16	17	18	19																																																									
20	21	22	23	24	25	26																																																									
27	28	29	30	31	1	2																																																									
3	4	5	6	7	8	9																																																									
Povratak	<table border="1"> <thead> <tr> <th colspan="7">januar 2022.</th> </tr> <tr> <th>pon</th><th>uto</th><th>sre</th><th>čet</th><th>pet</th><th>sub</th><th>ned</th> </tr> </thead> <tbody> <tr><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr> <tr><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td></tr> <tr><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td></tr> <tr><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr> <tr><td>31</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr> </tbody> </table> <p>Today: 8.9.2021.</p>						januar 2022.							pon	uto	sre	čet	pet	sub	ned	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	
januar 2022.																																																															
pon	uto	sre	čet	pet	sub	ned																																																									
27	28	29	30	31	1	2																																																									
3	4	5	6	7	8	9																																																									
10	11	12	13	14	15	16																																																									
17	18	19	20	21	22	23																																																									
24	25	26	27	28	29	30																																																									
31	1	2	3	4	5	6																																																									
<b>Ukupni kapaciteti: 200 soba</b>							<b>Dostupni kapaciteti: 171</b>																																																								

На трећој страници која се отвара кликом на дугме **Tutorial** на почетној форми, налази се кратка пратећа техничка документација са описом структуре и функционалности апликације. Четврто дугме (**Exit**) на почетној форми, затвара целу апликацију.

Базу треба проширити тако да се задовоље потребе наручиоца. За сваку уплату, потребно је памтити и датум када је уплата извршена.

Такође, хотел жели да подигне квалитет својих услуга и да понуди за сваку собу додатне погодности (интернет, телевизор, мини бар...). У свакој од соба постоји могућност увођења већег броја додатака који се могу одликовати одређеним специфичностима (нпр. више соба имају излаз на интернет, али брзина протока није свуда иста, тип телевизора... )

Креирати дијаграм и бекап стања проширене базе.

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ–А27

### НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Осигурање возила

Осигуравајућа кућа води евиденцију о продатим полисама осигурања. Поред података о продатим полисама, у бази се памте и подаци о возачима-власницима аутомобила, као и о моделима осигураних возила. За различите моделе аутомобила памте се и слике. Слика је запамћена у самој бази као податак бинарног типа.

За део апликације који памти податке о моделима аутомобила, потребно је омогућити додавање слике за моделе где слика није доступна, као и брисање погрешно постављене слике.

У делу апликације која памти податке о полисама, потребно је додати функционалност приказа полиса које ће истећи у неком наредном периоду, у циљу слања понуда за продужење полисе.

У циљу вођења прецизне евиденције о уплатама и

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

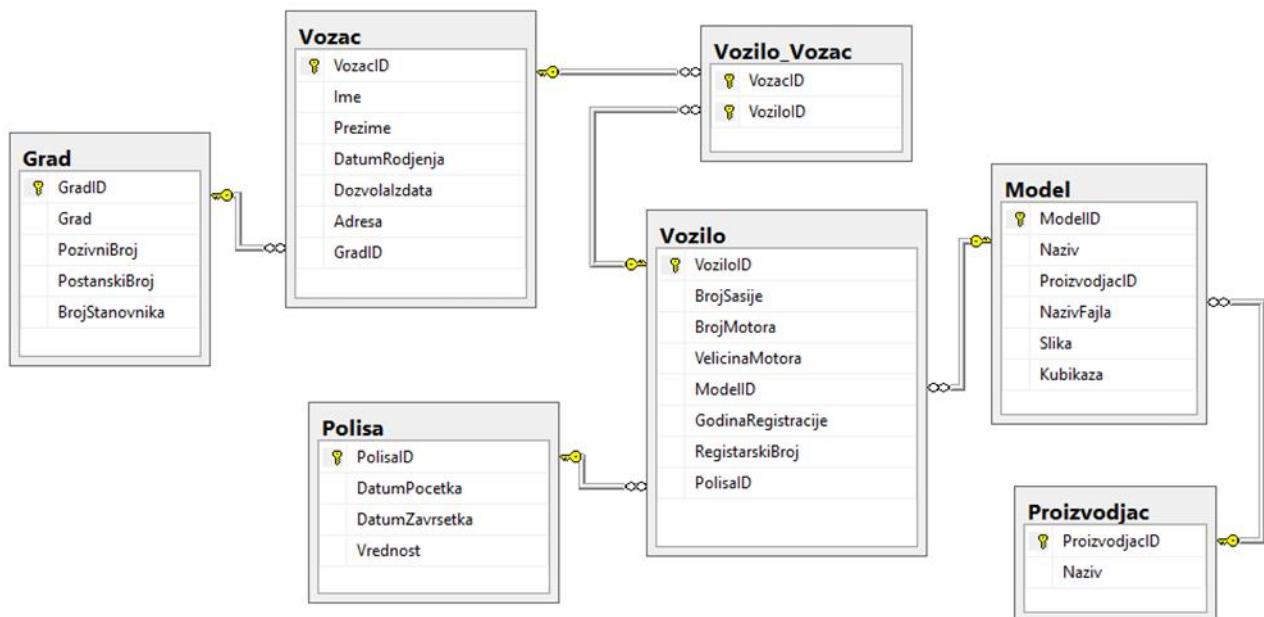
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A271**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A272**.

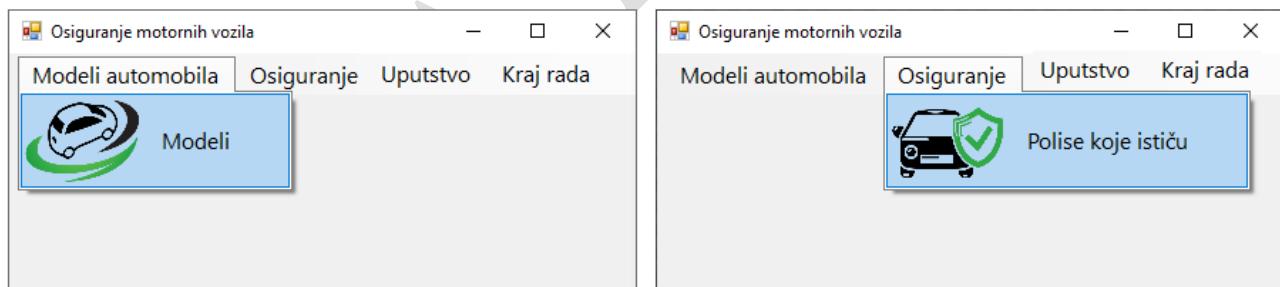
Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:

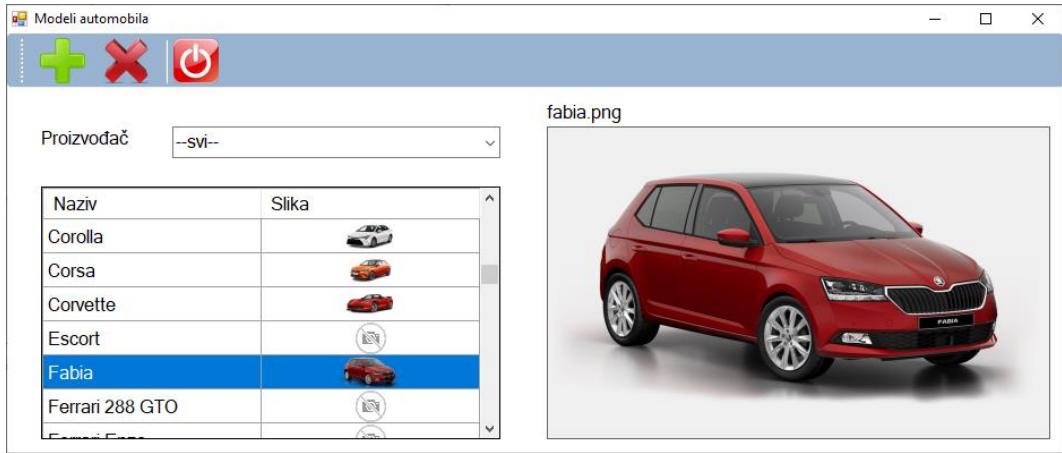


Развојним планом апликације предвиђено је да се уради следеће:

Креирати стартну форму која има мени са три команде. У менију **Модели аутомобила** налази се команда која отвара форму за ажурирање података о моделима аутомобила постављањем и брисањем слике модела.

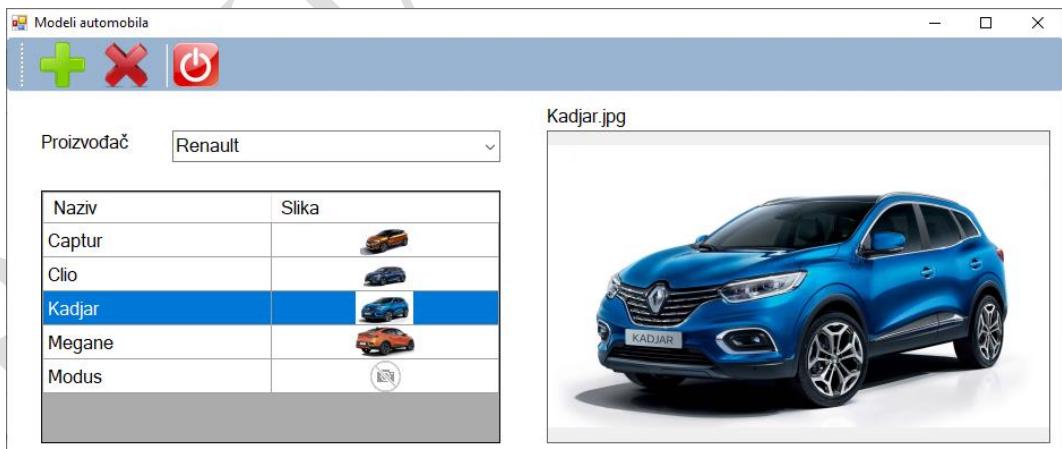
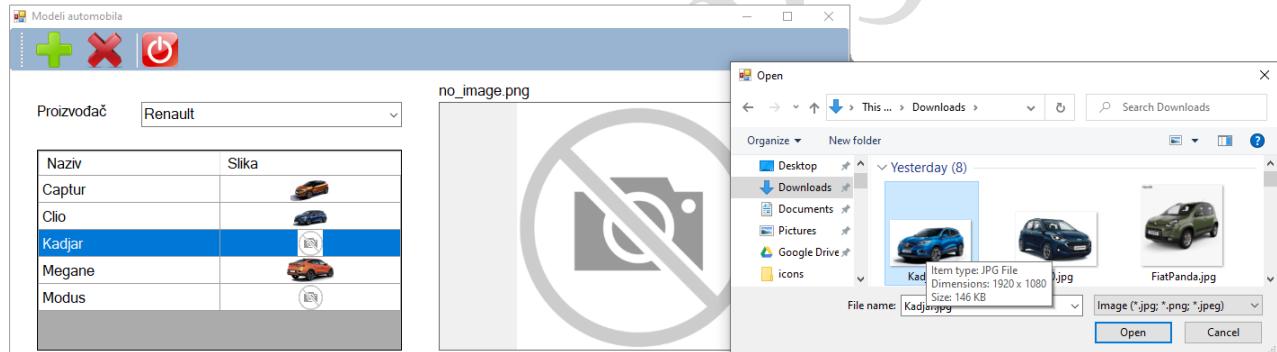


У падајућој листи **Произвођач** налазе се произвођачи аутомобила сортирани по алфабетном поретку. По учитавању у листи је избрана ставка **--сви--** чиме су у **dataGridView** контроли приказани сви модели, свих производача. У циљу лакше претраге, избором у листи производача, приказују се само модели изабраног производача.



Кликом на модел у списку модела, приказује се његова слика у **pictureBox**-у. Ако слика модела није доступна, приказује се подразумевана слика **no\_image.png**.

Прва команда у линији са алатима отвара дијалог за избор слике. Изабрана слика се приказује, а њен назив се исписује изнад **pictureBox**-а и ажурирају се подаци о моделу у бази. После ажурирања садржаја у бази, освежава се приказ у **dataGridView** контроли, а у случају грешке избацију се упозорење.



Друга команда брише погрешно уметнуту слику и враћа подразумевану слику **no\_image.png**. Трећа команда у линији са алатима затвара форму, али не и целу апликацију.

Ставка **Осигурање** на главном менију почетне форме има подмени са командом која отвара форму за преглед података о полисама које су близу истека.

Polise blizu roka isteka

Ime	Prezime	Model	Proizvođač	Registracija	Broj polise	Važi od	Važi do
Leon	Dimitrijević	Fiat Panda	Fiat	KA-280-LO	713	12.08.2020	11.09.2021
Jana	Branković	Touareg	BMW	JF-633-QI	738	13.03.2021	11.09.2021
Vesna	Andrejić	Accent	Hyundai	CX-545-OX	494	28.08.2020	14.09.2021
Stojanka	Bajić	Fiat Bravo/...	Fiat	SZ-182-OO	454	19.09.2020	14.09.2021
Žana	Avramović	Volvo Truck	Volvo	HE-946-CT	475	07.03.2021	15.09.2021
Danijal	Filipović	Kadjar	Renault	CX-385-HK	210	01.02.2021	15.09.2021
Nemanja	Barjaktare...	Focus	Ford	IG-795-AW	131	12.10.2020	19.09.2021
Uroš	Bebić	Volvo Penta	Volvo	IU-982-VW	131	12.10.2020	19.09.2021

Datum: 10.09.2021

Могуће је изабрати временски период унутар кога се тражи преглед стања полиса. Уколико период није изабран, подразумевано се приказују полисе које ће истећи у наредних 7 дана. Преглед се приказује кликом на дугме **Прикажи**. Дугме **Изађи** затвара форму.

На трећој страници која се отвара кликом на команду **Упутство** на почетној форми, налази се кратка пратећа техничка документација са описом структуре и функционалности апликације. Четврта команда (**Крај рада**) на менију почетне форме, затвара целу апликацију.

Базу треба проширити тако да се задовоље потребе наручиоца. 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 дијаграм и бекап стања проширене базе.

НАЗИВ РАДНОГ ЗАДАТКА: **Израда десктоп апликације: Стовариште материјала**

Трговинско предузеће тргује грађевинским материјалом и електро опремом. У циљу вођења евиденције о расположивој роби, као и о оствареним продајама креирана је база над њом је потребно креирати одговарајућу апликацију.

За сваку остварену куповину генерише се поруџбеница са побројаним ставкама купљене робе: назив производа, количина и укупан износ ставке у моменту куповине. При куповини се памти датум куповине-поручивања. Када роба буде испоручена купцу, у одговарајуће поље се уписује датум извршења испоруке (до тада ово поље је празно).

Све до момента испоруке, могуће је додати нове ставке на поруџбеницу. Оног момента када роба са конкретне поруџбенице буде испоручена, није могућа њена даља корекција, већ је потребно креирати нову поруџбеницу и започети нову куповину.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулацију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_1**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

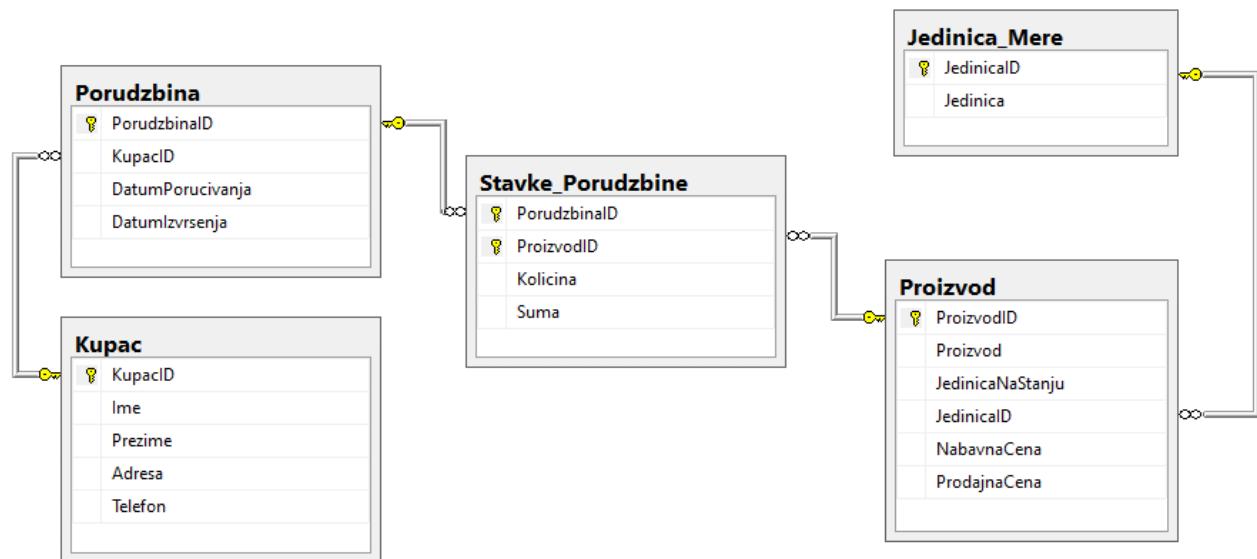
Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Дијаграм почетне структуре базе дат је као **Прилог\_4ЕИТ\_A281**.

Резервна копија почетног стања базе дата је као **Прилог\_4ЕИТ\_A282**.

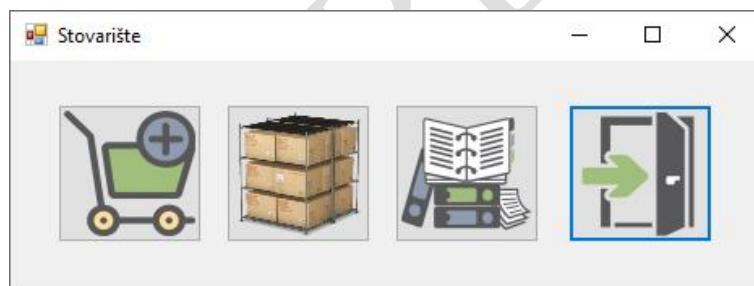
Архива икона дата је као **Прилог\_4ЕИТ\_A00**.

## ПРИЛОГ:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати стартну форму са четири тастера. Први тастер отвара форму на којој се у већ креiranу поруџбеницу могу убацити додатне ставке за куповину.



У текст поље се уноси шифра поруџбине коју треба изменити и кликом на дугме са лупом, врши се претрага и приказ података о поруџбеници. Ако је роба испоручена, приказује се порука да промена није могућа и дугме за убаџивање у корпу је затворено.

**Unos stavki porudžbine**

Šifra	Naziv proizvoda	Jedinica mere	Cena (RSD)	Izabrani proizvodi	Količina
64	Rizla bela 1	kilogram	11939,11	<input type="checkbox"/>	
71	Masa za fuqovanje ...	kilogram	2690,87	<input type="checkbox"/>	
79	Akrilin pripremljen kit	kilogram	3964,99	<input type="checkbox"/>	
100	Rizla bela 2	kilogram	6382,76	<input type="checkbox"/>	
113	Glet masa 20/1	kilogram	11117,84	<input type="checkbox"/>	

Dodaj u korpu      Zatvori

**Unos stavki porudžbine**

Šifra	Naziv proizvoda	Jedinica mere	Cena (RSD)	Izabrani proizvodi	Količina
11	Boja za beton bela	litar	11604,81	<input type="checkbox"/>	
12	Brusna ploča metal...	komad	6184,74	<input type="checkbox"/>	
13	Dimnjacka obloga ...	komad	7636,33	<input type="checkbox"/>	
14	Antenski priključak	komad	4519,47	<input type="checkbox"/>	
18	Burgija 10x210mm	komad	1740,63	<input type="checkbox"/>	

Dodaj u korpu      Zatvori

Ако роба није испоручена, могуће је додати нове ставке за куповину.

Листа производа приказује све расположиве производе. У листи производа се прво чекира производ који треба додати у корпу, а потом се унесе количина. Кликом на **Додај у корпу**, у случају успешног уписа нових ставки куповине приказује се порука и нови садржај корпе. Ако куповина није успела, приказује се обавештење са одговарајућом поруком.

**Unos stavki porudžbine**

Šifra porudžbine	Kupovina
69	Aleksa Filipović Datum poručivanja: 03.05.2021
Sadržaj korpe:	
1: Poludisperzija 25/1 A, količina: 5, iznos: 48473,80RSD	
2: Armiračka klešta, količina: 5, iznos: 14188,98RSD	
3: Boja za beton bela, količina: 3, iznos: 34814,43RSD	
4: Boja nitro emajl srebrna, količina: 2, iznos: 10721,54RSD	
5: Masa za fuqovanje parketa, količina: 13, iznos: 41674,31RSD	

**Pretraži**

Šifra	Naziv proizvoda	Jedinica mere	Cena (RSD)	Izabrani proizvodi	Količina
30	Boja nitro emajl srebrna	litar	5360,77	<input checked="" type="checkbox"/>	2
3	Boja za beton braon	litar	11722,3	<input type="checkbox"/>	
7	Poludisperzija 25/1 A		76	<input checked="" type="checkbox"/>	5
8	Boja nitro emajl crna		05	<input type="checkbox"/>	
11	Boja za beton bela		4,81	<input checked="" type="checkbox"/>	3

**Uspesna porudžbina**

**Dodaj u korpu** **OK** **Zatvori**

Дугме **Затвори** затвара форму за куповину, али не и целу апликацију.

Друго дугме на почетној форми отвара преглед робе које нема у довољним количинама у магацину и постоји потреба за њеним требовањем.

За изабрани опсег датума претражују се све поруџбине реализоване у задатом периоду и одређује колика је укупна продаја за сваки од производа (укупно продатих комада тј јединица мере).

Уколико на залихама нема довољно производа за реализацију свих испорука за изабрани период продаје, потребно је извршити наручивање и у ту сврху треба у листи приказати све производе које треба наручити (назив производа, јединица мере, колико је расположиво у магацину и колико је продато и потребно за испоруку).

Преглед се приказује кликом на дугме **Прикажи**. Дугме **Изађи** затвара форму.

**Trebovanje robe**

Kupovina obavljena u periodu od-do:

1. 1.2021. 19. 9.2021.

Proizvodi kojih nema na zalihamama:

Proizvod	Jedinica mере	Na zalihamama	Za isporuku
Burgija 12x160mm	komad	27	40
AG77 emulzioni premaz na b...	litar	26	29
Baštensko crevo 25m 1/2"	komad	10	20
Betonski blok 25x20x40	komad	4	26

**Prikaži** **Izađi**

На трећој страници која се отвара кликом на команду **Упутство** са почетне форме, налази се кратка пратећа техничка документација са описом структуре и функционалности апликације. Четврта команда (**Крај рада**) на менију почетне форме, затвара целу апликацију.

Базу треба проширити тако да се задовоље додатне потребе наручиоца.

Потребно је регистровати и податке о добављачима од којих се производи набављају. За добављача треба да памтимо назив, адресу, град, број телефона и мејл адресу. Предвидети да се један производ може набављати од више различитих добављача. За производ је потребно памтити и набавну цену.

Продавац намерава да омогући да се поједине ставке са исте поруџбенице не морају испоручивати све истовремено, већ да могу бити испоручене временски независно једна од друге. Када буде испоручена и последња ставка, сматра се да је извршена комплетна испорука по поруџбеници. Због тога је потребно проширити базу и омогућити памтење датума испоруке за сваку појединачну ставку поруџбенице.

Креирати дијаграм и бекап стања проширене базе.

РАДНА ВЕРЗИЈА

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ- А29

НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Голф играчи

Спортска организација Голф клуб Србија води евиденцију о свим голф теренима и играчима на територији Србије. За потребе клуба креирана је база података у којој се води евиденција о свим играчима, теренима и одржаним партијама. Потребно је проширити постојећу апликацију са додавањем функционалности уписа података за нове играче , као и за приказ колико је партија одржано по сваком терену у одређеном периоду. Такође је потребно проширити базу са подацима о тренерима и тренинзима, како би се водила евиденција о тренирању и припреми играча за мечеве. Креирати десктоп апликацију која ће омогућити рад са постојећом базом.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулатију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_29**, где се као име и презиме уноси име кандидата.

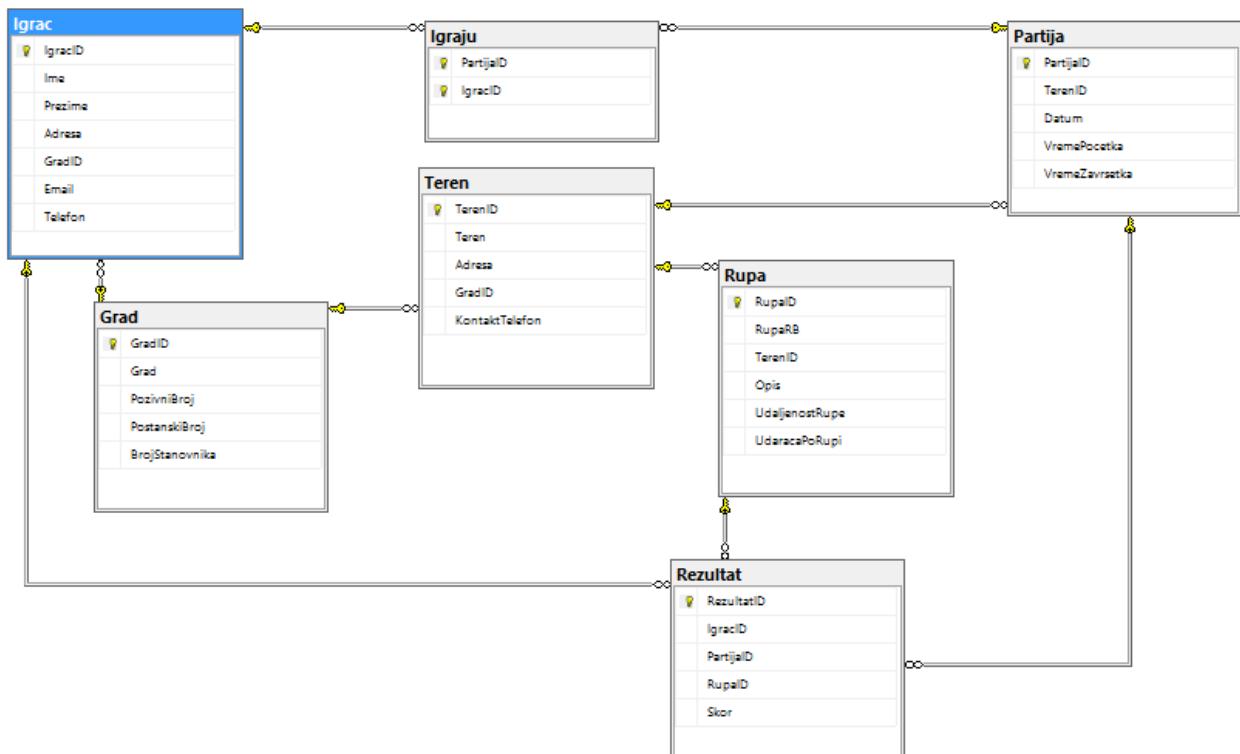
Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка кандидат може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A29\ Потребни ресурси**.

## Прилог за задатак:



Развојним планом апликације предвиђено је да се уради следеће:

Креирати форму која податке о свим играчима приказује у **listBox** контроли. Приказани подаци треба да буду сортирани по шифри. Након селекције реда у **listBox** контроли или променом шифре у пољу **Šifra** на форми, подаци о играчу се приказују у одговарајућим пољима форме. За приказ поља град користити **comboBox** контролу. **ComboBox** контролу иницијално треба попунити са именима градова, који су сортирани у растућем поретку.

Омогућити кориснику унос новог играча. Приликом уписа играча у базу аутоматски треба да се генерише вредност шифре. Припрема форме за унос омогућава се кликом на дугме **Priprema za unos**. Под припремом се подразумева да садржај свих појединачних поља буде очишћен и да поље за унос шифре има само дозволу приступа за читање. Играч се може уписати тек након активираног думета **Priprema za unos** и попуњених свих појединачних поља. Након уноса ажурирати садржај **listBox** контроле. Поруку о успешности уноса приказати у **MessageBox**-у.

Функционалност менија:

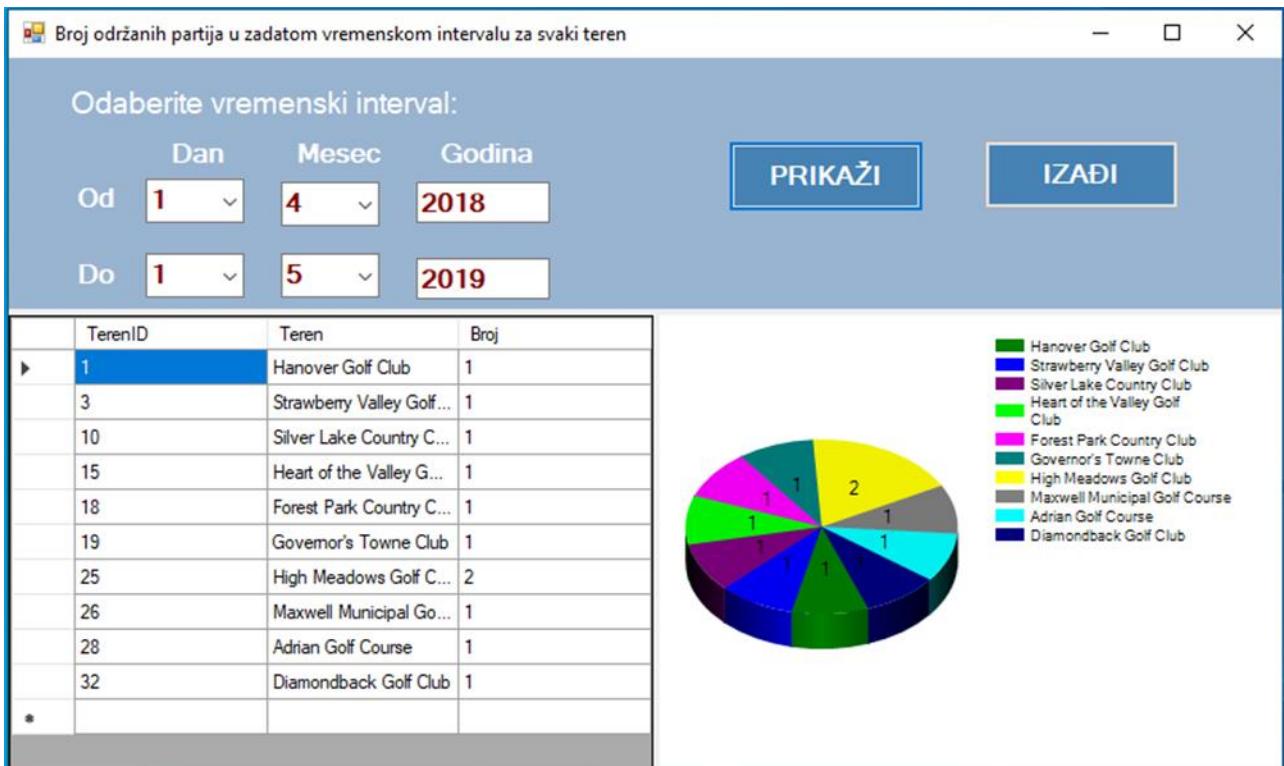
- Ставка менија **Unesi novog igrača** омогућава унос података новог играча;
- Ставка менија **Analiza** приказује форму са графиконом;
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство;
- Ставка менија **Izlaz** затвара апликацију.

Podaci selektovanog igrača:							
	ID	Ime	Prezime	Adresa	GradID	Mejl	Telefon
Šifra	1	Bartlett	Nicole	Plymouth, Bude 78	41	taeuyu@yahoo.net	068-264-4016
Ime	Nicole						
Prezime	Bartlett						
Adresa	Plymouth, Bude 78						
Grad	Boston						
Mejl	taeuyu@yahoo.net						
Telefon	068-264-4016						
<input type="button" value="Priprema za unos"/>							

Unesite podatke o novom igraču:																																																																																																																																																																									
	ID	Ime	Prezime	Adresa	GradID	Mejl	Telefon																																																																																																																																																																		
<b>Šifra</b>																																																																																																																																																																									
<b>Ime</b>																																																																																																																																																																									
<b>Prezime</b>																																																																																																																																																																									
<b>Adresa</b>																																																																																																																																																																									
<b>Grad</b>	Izaberite grad...																																																																																																																																																																								
<b>Mejl</b>																																																																																																																																																																									
<b>Telefon</b>																																																																																																																																																																									
				Priprema za unos	Odustani																																																																																																																																																																				
<table border="1"> <thead> <tr> <th></th> <th>ID</th> <th>Ime</th> <th>Prezime</th> <th>Adresa</th> <th>GradID</th> <th>Mejl</th> <th>Telefon</th> </tr> </thead> <tbody> <tr><td>1</td><td>Nicole</td><td>Bartlett</td><td>Plymouth, Bude 78</td><td>41</td><td>taeuuy@yahoo.net</td><td>068-264-4016</td></tr> <tr><td>2</td><td>Christine</td><td>Rubio</td><td>Liverpool, Banff 68</td><td>28</td><td>vavgr.tohx@mail.com</td><td>079-855-6508</td></tr> <tr><td>3</td><td>Pablo</td><td>Werner</td><td>St Helens, Ulloapool 678</td><td>28</td><td>guubm.jkqo@yahoo.com</td><td>067-064-4824</td></tr> <tr><td>4</td><td>Darnell</td><td>Calderon</td><td>Preston, Lochgoilhead 5</td><td>23</td><td>ws121@outlook.org</td><td>078-284-9875</td></tr> <tr><td>5</td><td>Desiree</td><td>Farmer</td><td>St Helens, Denbigh 6</td><td>14</td><td>mldg.qlerx@mail.com</td><td>076-134-9872</td></tr> <tr><td>6</td><td>Holly</td><td>Fernandez</td><td>Watford, Forest Hill 912</td><td>24</td><td>tgwx.hnyr@gmail.com</td><td>086-611-0505</td></tr> <tr><td>7</td><td>Chadwick</td><td>Ferraro</td><td>Bristol, Gloucester 10</td><td>24</td><td>louisw.schaeffer@mail.com</td><td>099-128-2816</td></tr> <tr><td>8</td><td>Julie</td><td>Moreno</td><td>Peterborough, Craven Arms 732</td><td>2</td><td>ynxad.jbog@gmail.net</td><td></td></tr> <tr><td>9</td><td>Daphne</td><td>Dudley</td><td>Bristol, Waltham Abbey 754</td><td>50</td><td>zebd@yahoo.com</td><td>088-304-5567</td></tr> <tr><td>10</td><td>Jean</td><td>BBB</td><td>Trevino</td><td>50</td><td>lbmc0@hotmail.net</td><td>066-785-2504</td></tr> <tr><td>11</td><td>Sonya</td><td>Ryan</td><td>Peterborough, Mablethorpe 7</td><td>43</td><td>xkjtz@gmail.org</td><td>086-999-4421</td></tr> <tr><td>12</td><td>Lindsay</td><td>Conner</td><td>Peterborough, Huddersfield 84</td><td>2</td><td>itivpg@mail.net</td><td>098-169-1104</td></tr> <tr><td>13</td><td>Katrina</td><td>Shaver</td><td>Gloucester, Bridport 584</td><td>27</td><td>gjvlyqgutlook@gmail.com</td><td>076-383-4177</td></tr> <tr><td>14</td><td>Leanne</td><td>Hanson</td><td>Oxford, North End 541</td><td>35</td><td>as1mre@gmail.com</td><td>086-311-5987</td></tr> <tr><td>15</td><td>Kari</td><td>Hunter</td><td>Nottingham, Bicester 423</td><td>23</td><td>uxiaman@hotmail.net</td><td>086-672-5666</td></tr> <tr><td>16</td><td>Chadwick</td><td>Velazquez</td><td>Northampton, Littlehampton 57950</td><td>23</td><td>uhua@hotmail.com</td><td>089-037-2714</td></tr> <tr><td>17</td><td>Seth</td><td>Gordon</td><td>Poole, Tresco 7</td><td>39</td><td>novq.icvx@yahoo.com</td><td>079-871-4208</td></tr> <tr><td>18</td><td>Warren</td><td>west</td><td>Blackburn, Sydenham 99</td><td>20</td><td>zfszu.nfxje@outlook.com</td><td>097-951-0768</td></tr> <tr><td>19</td><td>Bernard</td><td>Rowe</td><td>Oxford, Swaffham 4</td><td>48</td><td>ucgw@mail.org</td><td>097-325-3632</td></tr> <tr><td>20</td><td>Shelby</td><td>Bradford</td><td>Nottingham, Stansted 934</td><td>36</td><td>mcd1@mail.org</td><td>077-561-6172</td></tr> <tr><td>21</td><td>Devon</td><td>Henry</td><td>Sheffield, Menai Bridge 6</td><td>14</td><td>uuwy@gmail.com</td><td>068-594-6941</td></tr> <tr><td>22</td><td>Tonia</td><td>Houston</td><td></td><td></td><td>kqsd1j@outlook.org</td><td>068-744-4084</td></tr> </tbody> </table>									ID	Ime	Prezime	Adresa	GradID	Mejl	Telefon	1	Nicole	Bartlett	Plymouth, Bude 78	41	taeuuy@yahoo.net	068-264-4016	2	Christine	Rubio	Liverpool, Banff 68	28	vavgr.tohx@mail.com	079-855-6508	3	Pablo	Werner	St Helens, Ulloapool 678	28	guubm.jkqo@yahoo.com	067-064-4824	4	Darnell	Calderon	Preston, Lochgoilhead 5	23	ws121@outlook.org	078-284-9875	5	Desiree	Farmer	St Helens, Denbigh 6	14	mldg.qlerx@mail.com	076-134-9872	6	Holly	Fernandez	Watford, Forest Hill 912	24	tgwx.hnyr@gmail.com	086-611-0505	7	Chadwick	Ferraro	Bristol, Gloucester 10	24	louisw.schaeffer@mail.com	099-128-2816	8	Julie	Moreno	Peterborough, Craven Arms 732	2	ynxad.jbog@gmail.net		9	Daphne	Dudley	Bristol, Waltham Abbey 754	50	zebd@yahoo.com	088-304-5567	10	Jean	BBB	Trevino	50	lbmc0@hotmail.net	066-785-2504	11	Sonya	Ryan	Peterborough, Mablethorpe 7	43	xkjtz@gmail.org	086-999-4421	12	Lindsay	Conner	Peterborough, Huddersfield 84	2	itivpg@mail.net	098-169-1104	13	Katrina	Shaver	Gloucester, Bridport 584	27	gjvlyqgutlook@gmail.com	076-383-4177	14	Leanne	Hanson	Oxford, North End 541	35	as1mre@gmail.com	086-311-5987	15	Kari	Hunter	Nottingham, Bicester 423	23	uxiaman@hotmail.net	086-672-5666	16	Chadwick	Velazquez	Northampton, Littlehampton 57950	23	uhua@hotmail.com	089-037-2714	17	Seth	Gordon	Poole, Tresco 7	39	novq.icvx@yahoo.com	079-871-4208	18	Warren	west	Blackburn, Sydenham 99	20	zfszu.nfxje@outlook.com	097-951-0768	19	Bernard	Rowe	Oxford, Swaffham 4	48	ucgw@mail.org	097-325-3632	20	Shelby	Bradford	Nottingham, Stansted 934	36	mcd1@mail.org	077-561-6172	21	Devon	Henry	Sheffield, Menai Bridge 6	14	uuwy@gmail.com	068-594-6941	22	Tonia	Houston			kqsd1j@outlook.org	068-744-4084
	ID	Ime	Prezime	Adresa	GradID	Mejl	Telefon																																																																																																																																																																		
1	Nicole	Bartlett	Plymouth, Bude 78	41	taeuuy@yahoo.net	068-264-4016																																																																																																																																																																			
2	Christine	Rubio	Liverpool, Banff 68	28	vavgr.tohx@mail.com	079-855-6508																																																																																																																																																																			
3	Pablo	Werner	St Helens, Ulloapool 678	28	guubm.jkqo@yahoo.com	067-064-4824																																																																																																																																																																			
4	Darnell	Calderon	Preston, Lochgoilhead 5	23	ws121@outlook.org	078-284-9875																																																																																																																																																																			
5	Desiree	Farmer	St Helens, Denbigh 6	14	mldg.qlerx@mail.com	076-134-9872																																																																																																																																																																			
6	Holly	Fernandez	Watford, Forest Hill 912	24	tgwx.hnyr@gmail.com	086-611-0505																																																																																																																																																																			
7	Chadwick	Ferraro	Bristol, Gloucester 10	24	louisw.schaeffer@mail.com	099-128-2816																																																																																																																																																																			
8	Julie	Moreno	Peterborough, Craven Arms 732	2	ynxad.jbog@gmail.net																																																																																																																																																																				
9	Daphne	Dudley	Bristol, Waltham Abbey 754	50	zebd@yahoo.com	088-304-5567																																																																																																																																																																			
10	Jean	BBB	Trevino	50	lbmc0@hotmail.net	066-785-2504																																																																																																																																																																			
11	Sonya	Ryan	Peterborough, Mablethorpe 7	43	xkjtz@gmail.org	086-999-4421																																																																																																																																																																			
12	Lindsay	Conner	Peterborough, Huddersfield 84	2	itivpg@mail.net	098-169-1104																																																																																																																																																																			
13	Katrina	Shaver	Gloucester, Bridport 584	27	gjvlyqgutlook@gmail.com	076-383-4177																																																																																																																																																																			
14	Leanne	Hanson	Oxford, North End 541	35	as1mre@gmail.com	086-311-5987																																																																																																																																																																			
15	Kari	Hunter	Nottingham, Bicester 423	23	uxiaman@hotmail.net	086-672-5666																																																																																																																																																																			
16	Chadwick	Velazquez	Northampton, Littlehampton 57950	23	uhua@hotmail.com	089-037-2714																																																																																																																																																																			
17	Seth	Gordon	Poole, Tresco 7	39	novq.icvx@yahoo.com	079-871-4208																																																																																																																																																																			
18	Warren	west	Blackburn, Sydenham 99	20	zfszu.nfxje@outlook.com	097-951-0768																																																																																																																																																																			
19	Bernard	Rowe	Oxford, Swaffham 4	48	ucgw@mail.org	097-325-3632																																																																																																																																																																			
20	Shelby	Bradford	Nottingham, Stansted 934	36	mcd1@mail.org	077-561-6172																																																																																																																																																																			
21	Devon	Henry	Sheffield, Menai Bridge 6	14	uuwy@gmail.com	068-594-6941																																																																																																																																																																			
22	Tonia	Houston			kqsd1j@outlook.org	068-744-4084																																																																																																																																																																			

Ставка менија ***Analiza*** отвара форму за табеларни приказ списка терена (шифра, назив) и број партија које су одигране на сваком терену у траженом временском интервалу. Приказан списак терена је сортиран по шифри терена у растућем редоследу. Кликом на дугме ***Prikaži*** исцртава се график, који приказује број партија на сваком терену у траженом временском интервалу. Поред графичког приказа потребно је податке приказати и у табели. Дугме ***Izađi*** затвара отворен образац, и враћа на почетни мени.

Образац треба да изгледа као на слици:



Осмислите проширење базе којим се омогућава чување података о тренерима и тренинзима. За сваког тренера желимо да чувамо следеће податке: име, презиме, адреса, телефон, email. Тренер води једну или више група. Група садржи следеће податке: назив групе, датум и време тренинга. Група може имати само један тренинг недељно. Групу чини више играча а терен се бира по препоруци тренера. Такође је потребно табелу **Teren** проширити са пољем категорија терена.

Креирати дијаграм и бекап стања проширене базе.

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-А20

НАЗИВ РАДНОГ ЗАДАТКА: Израда десктоп апликације: Голф терени

Спортска организација Голф клуб Србија води евиденцију о свим голф теренима и играчима на територији Србије. За потребе клуба креирана је база података у којој се води евиденција о свим играчима, теренима и одржаним партијама. Потребно је проширити постојећу апликацију са додавањем функционалности брисања терена који више нису актуелни, као и за приказ партија које су трајале у оквиру изабраног временског интервала. За потребне такмичења потребно је проширити постојећу базу са подацима о праћењу судија по одиграним партијама. Креирати десктоп апликацију која ће омогућити рад са постојећом базом.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати десктоп апликацију у циљу задовољења захтева;
- Повезати десктоп апликацију са базом података;
- Креирати упите за манипулатију подацима у бази података;
- Тестирати функционалности десктоп апликације;
- Модификовати структуру базе података у складу са захтевима корисника;
- Написати упутство корисницима о коришћењу апликације;
- ЕР дијаграм, бекап базе и апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_30**, где се као име и презиме уноси име ученика.

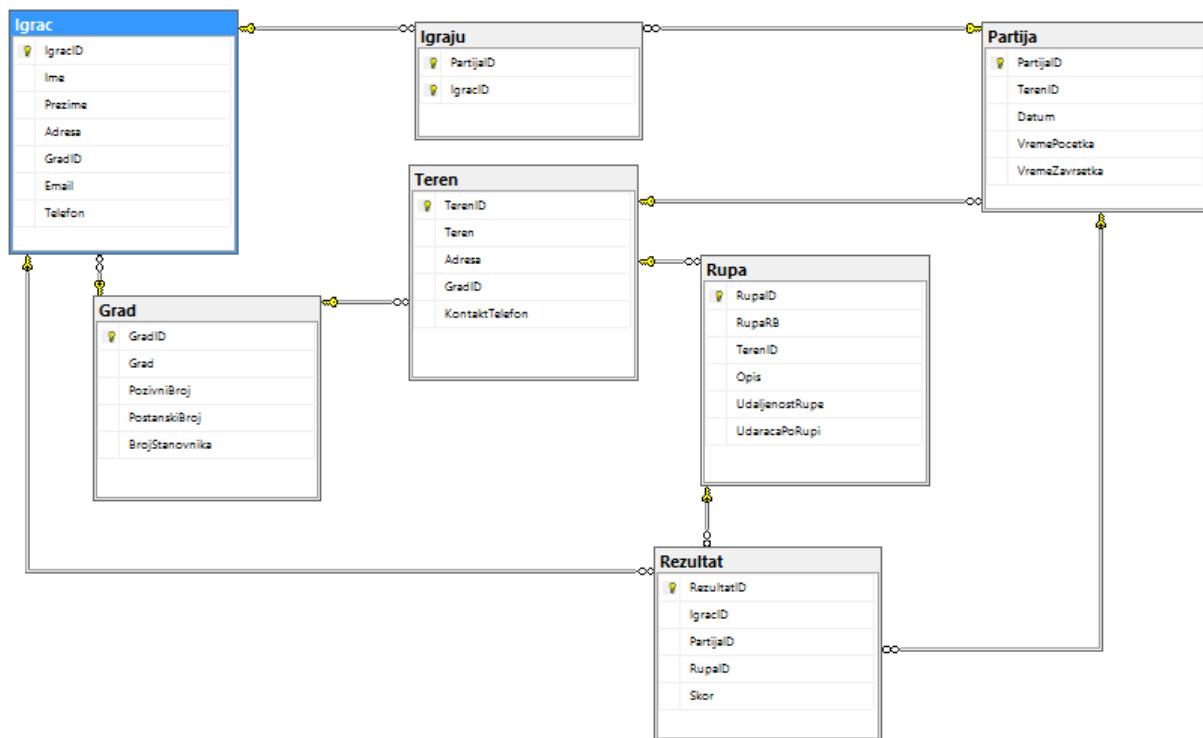
Предвиђено време за израду задатка је 180 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Дијаграм почетне структуре базе, резервна копија почетног стања базе и архива икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_A30\Потребни ресурси**.

## Прилог за задатак:



Развојним планом апликације предвиђено је да се уради следеће:

Креирана форма поред менија садржи и приказ података о свим голф теренима. Тражени подаци су приказани у контроли **listBox** (шифра, назив, адреса, телефон, град). Такође на форму треба поставити поља за појединачни приказ података за изабрани терен: шифра, назив терена, адреса, град и телефон. За приказ поља град користити **comboBox** контролу. **ComboBox** контролу иницијално треба попунити са именима градова. Појединачна поља форме иницијално треба попунити са подацима терена који се налази у првој врсти табеле која је сортирана по шифри у растућем поретку. Након селекције реда у **listBox** контроли, подаци о терену се приказују у појединачним пољима форме. Такође, треба омогућити да се након уноса шифре у поље за шифру, подаци о терену приказују у одговарајућим пољима форме, а ако не постоји терен са унетом шифром кориснику исписати поруку: „Нема података за терен са унетом шифром.“. Кликом да дугме менија **Obriši**, из базе података брише се терен чије су вредности приказане у појединачним пољима форме. Пре извршене акције брисања корисник треба да потврди да је сигуран да жели да брише одабран терен. Након брисања податка у бази, освежити и приказ у **listBox** контроли. Изглед форме је дат на слици која у заглављу има назив „Evidencija golf terena“. Поруку о успешности брисања приказати у **MessageBox**-у.

Функционалност менија:

- Ставка менија **Obriši** омогућава брисање података о терену који је затворен;
- Ставка менија **Analiza** приказује форму са графиконом;
- Ставка менија **O aplikaciji** треба да отвори кратко корисничко упутство;
- Ставка менија **Izlaz** затвара апликацију.

Evidencija golf terena

Šifra	1
Naziv	Hanover Golf Club
Adresa	Tampa 741
Grad	Fremont
Telefon	096-925-5372

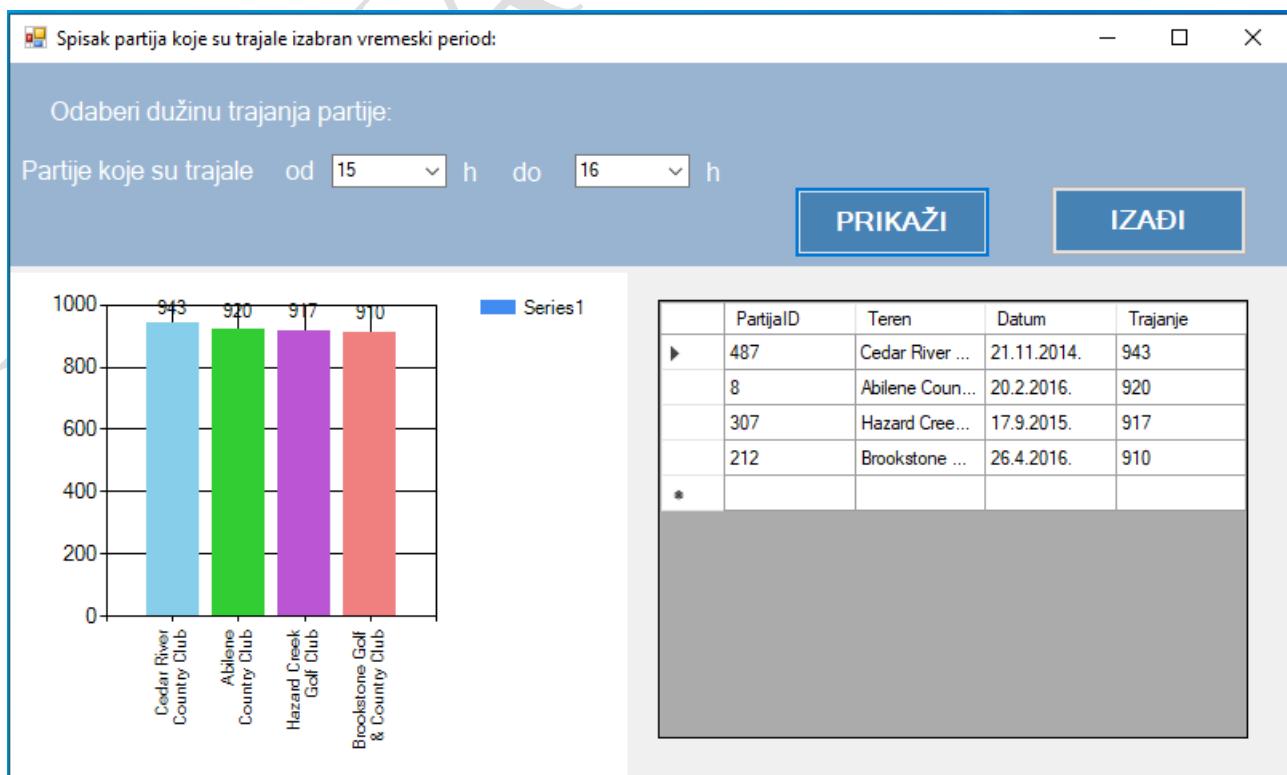
ID	Naziv	Adresa	Telefon	Grad
1	Hanover Golf Club	Tampa 741	096-925-5372	Fremont
2	Abilene Country Club	Mesa 84	097-271-7108	Boston
3	Strawberry Valley Golf Club	Arlington 779	099-748-5132	Jersey
4	Cedar River Country Club	New York 451	066-276-3907	Lubbock
5	Boulder Creek Golf Club	Dallas 772	098-702-7351	Chicago
6	The Wetlands Golf Club	Philadelphia 18	086-487-6289	Lubbock
7	Grays Harbor Golf Club	Jacksonville 24	078-504-7929	Newark
8	The Bridges Golf Club	Raleigh 721	076-994-6027	Columbus
9	Ruggles Golf Club	Lubbock 77	088-201-9604	Jacksonvil
10	Silver Lake Country Club	El Paso 83	079-823-2141	Nashville

Obriši  
 Analiza  
 O aplikaciji  
 Izlaz

Ставка менија **Analiza**, табеларно 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.

Дугме **Prikaži** исцртава график који приказује време трајања партија на сваком терену у траженом временском интервалу. Дугме **Izadi** затвара отворен образац, и враћа на почетни мени.

Образац треба да изгледа као на слици:



Осмислите проширење базе којим се омогућава чување података о судијама. За сваког судију желимо да чувамо следеће податке: име, презиме, адреса, телефон, емаил. Судија може да суди на више терена а на терену може да буде више судија. Такође је потребно табелу **Teren** проширити са пољем категорија терена.

Креирати дијаграм и бекап стања проширене базе.

РАДНА ВЕРЗИЈА

## РАДНИ ЗАДАЦИ - КОМПЕТЕНЦИЈА Б

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б1

НАЗИВ РАДНОГ ЗАДАТКА: **Фото галерија**

У склопу припрема за предстојећи конгрес на тему Старог века, департман за историју Филозофског факултета, исказао је потребу за креирањем веб страница са прегледом и основним информацијама на тему седам светских чуда античког света.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_2**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Текстуални фајл са описом слика дат је као **Прилог\_4ЕИТ\_B11**

Архива слика и икона дата је као **Прилог\_4ЕИТ\_B00**.

## ПРИЛОГ:

Креирати веб апликацију са називом **Фото галерија** која се састоји од три странице: **Почетна, О аутору и Упутство.**

На страницу **Почетна** поставити контејнер за смештај галерије величине 640x480px. На средину контејнера поставити централну фотографију величине 320x200px, а испод ње распоредити десет малих фотографија величине 64x40px. Изнад централне фотографије приказати текстуални опис фотографије који се чита из текстуалног фајла датог у прилогу задатка. Описи су дати у следећем формату: за сваку слику унет је по један параграф који садржи **име фајла: опис слике**

Фотографије се учитавају из посебног фолдера и произвољних су димензија, а урађене су у **JPG** формату. Фотографије имају називе *slika1.jpg*, *slika2.jpg*, ... до *slika10.jpg*.

На страници **О аутору** написати основне податке о аутору.

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 опис 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.

## Фото галерија

Почетна    О аутору    Упутство    Завршни испит

Седам чуда се први пут помињу око 450. пре н. е., када је грчки историчар Херодот саставио списак грађевина којима су се дивили, а међу њима су биле пирамиде у Гизи и Висећи вртovi. Исто су учинили и византијски филозоф Филон, око 225. пре н. е., и грчки писац Антипатер из Сидона, око 130. пре н. е. На њиховим пописима први пут је заједно споменуто свих седам грађевина које су данас познате као седам светских чуда старог света. Стари Грци су тај список назвали "Седам атракција насељене Земље које треба обавезно видети" и односе се на скуп архитектонских дела за која су сматрали врхунцем људске креативности и генијалности. Од свих тада изабраних чуда, данас једино можемо

Where in the  
Ancient World  
Are the  
Seven Wonders?

Rome    GREECE    CRETE    BLACK SEA    ASIA MINOR (TURKEY)    MESOPOTAMIA (IRAQ)    REDS

Ephesus    TEMPLE OF ARTEMIS    Halicarnassus    MAUSOLEUM AT HALICARNASSUS    Rhodes    COLOSSUS OF RHODES    Alexandria    PHAROS OF ALEXANDRIA    Giza    GREAT PYRAMID AT GIZA    Jerusalem    Babylon    HANGING GARDENS OF BABYLON

## Фото галерија

Почетна

О аутору

Упутство

Завршни испит

Вртови из Вавилона познати су још као "Семирамидини висећи вртови". Постоји мит да је краљ Набукодоносор око 600. године п. н. е. наредио да се изграде вртови на терасама палате у жељи да орасположи краљицу Семирамиду. Остаци висећих вртова у Вавилону налазе се на источној обали реке Еуфрат, на око 50 км јужно од Багдада, у Ираку. Детаљни описи висећих вртова пронађени су у делима грчких историчара Бероса и Диодора. Запамћујућа је чињеница да ипак не постоје докази да су вртови икада постојали, иако су у људској машти несумњиво проналазили пут до необичног и задивљујућег.



Трећа страница (**Упутство**) садржи кратко корисничко упутство и објашњење структуре апликације.

РАДНИ

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б2**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Домаће животиње**

У склопу припрема за нову школску годину, Предшколска установа “Срећно дете”, исказала је потребу за креирањем веб страница са темом упознавања деце у узрасту од две године са домаћим животињама и њиховим стаништем.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б2**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Слике животиња, звучни фајлови са карактеристичним звуцима животиња снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б2\**. Потребни ресурси.

## ПРИЛОГ:

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.

Фотографије се учитавају из посебног фолдера и произвољних су димензија, а урађене су у JPEG формату. Звучни фајлови се учитавају из посебног фолдера.

На страници **О аутору** написати основне податке о аутору.

Трећа страница (**Упутство**) садржи кратко корисничко упутство и објашњење структуре 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.



**ДОМАЋЕ ЖИВОТИЊЕ**

ПочетнаО ауторуУпутство



**ПАС**

Пас је најстарија домаћа животиња. Одана је човеку. Има добро развијено памћење и чуло мириза. Отпашава се пајањем.

Copyright © Предшколска Установа "Срећно дете"



РАДНА ВЕРЗИЈА

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б3

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Србија и њени суседи

У склопу припрема за нову школску годину, основна школа „Соња Маринковић“, исказала је потребу за креирањем веб странице са темом из географије: Србија и њени суседи.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б3**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Звучни фајлови са звуцима химни и архива слика и икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б3\ Потребни ресурси**.

РАДНА ПОВРШИНА

## ПРИЛОГ:

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.

Фотографије се учитавају из посебног фолдера и произвољних су димензија, а урађене су у JPG формату. Звучни фајлови химни се учитавају из посебног фолдера.

На страници **О аутору** написати основне податке о аутору.

Трећа страница (**Упутство**) садржи кратко корисничко упутство и објашњење структуре 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.

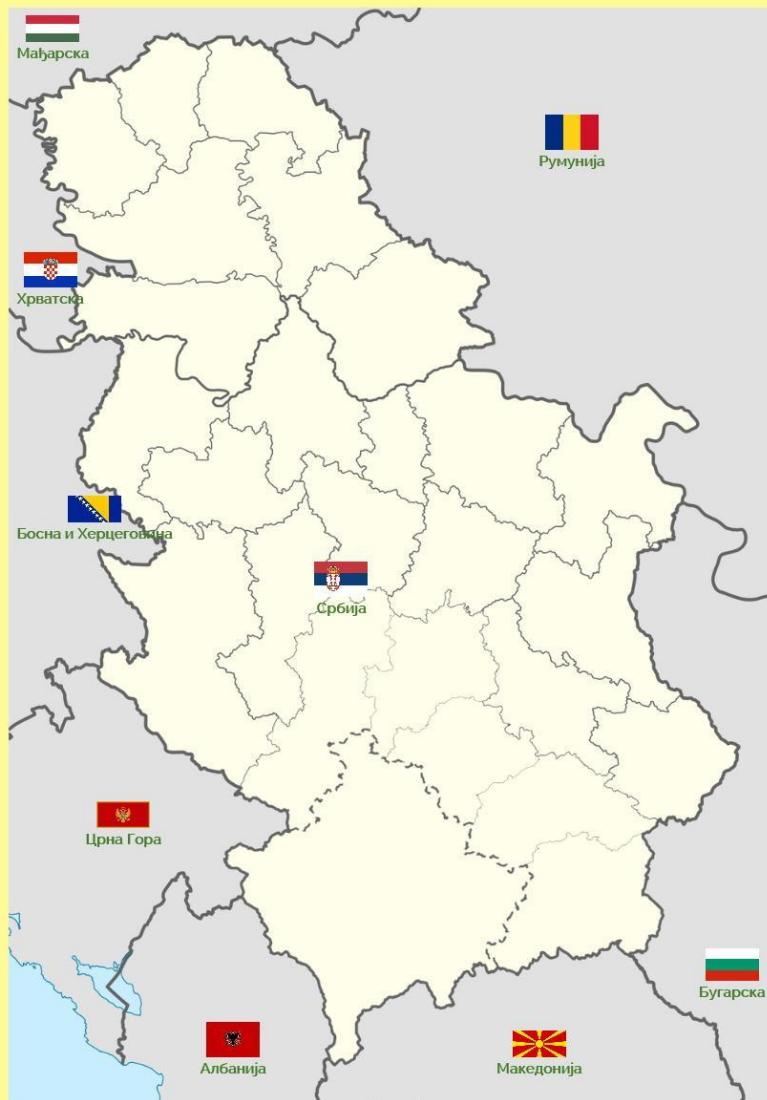
# ОСНОВНА ШКОЛА "СОЊА МАРИНКОВИЋ"

Почетна

О аутору

Упутство

## Србија и њени суседи



Copyright © Основна школа "Соња Маринковић"

# Географија

РАД

# ОСНОВНА ШКОЛА "СОЊА МАРИНКОВИЋ"

Почетна

О аутору

Упутство

## Србија и њени суседи



Copyright © Основна школа "Соња Маринковић"

РАД

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б4

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Слајдер за слике (Image slider)

У склопу припрема за нову школску годину, основна школа „Соња Маринковић“, исказала је потребу за креирањем веб странице са темом из ликовног: Пикасо.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б4**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно

Архива слика и икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б4\ Потребни ресурси**.

## ПРИЛОГ:

Дат је фолдер са сликама и следећа HTML структура на веб страници:

```
<div id="slike">

</div>
```

Креирати веб апликацију са називом **Пикасо** која се састоји од три странице: **Почетна**, **О аутору** и **Упутство**.

На страницу **Почетна** поставити 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 кружића 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. Такође, потребно је истовремено са сменом слика у слайдеру, у диву поред слике приказати и податке о активној слици (назив, годину, кратак опис и слично).

На страници **О аутору** написати основне податке о аутору (име, презиме и имејл ученика, име апликације и датум креирања).

Трећа страница (**Упутство**) садржи кратко корисничко упутство и објашњење структуре апликације.

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.

# ОСНОВНА ШКОЛА "СОЊА МАРИНКОВИЋ"

Почетна

О аутору

Упутство

П  
и  
к  
а  
с  
о



Фаун са звездама

1955

Фаун са звездама настало је 1955. године, на врхунцу Пикасове романсе са Жаклином Рок, са којом се оженио 1961. Декоративни и разиграни графички квалитет рада карактеристичан је за Пикасов касни стил и посебно се односи на осликану керамику коју је почeo израђивати 1947.



Copyright © Основна школа "Соња Маринковић"

РАДНА ВЕРСИЈА

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б5

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Звуци инструмената

У склопу припрема за нову школску годину, основна школа „Соња Маринковић“, исказала је потребу за креирањем веб странице за потребе предмета Музичка култура са темом : Музички инструменти.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_5**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Слике инструмената, звучни фајлови са звучима инструмената за најмање шест инструмената и архива слика и икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б5\ Потребни ресурси**.

### ПРИЛОГ:

Креирати веб апликацију са називом **Звуци инструмената** која се састоји од три странице: **Почетна**, **О аутору** и **Упутство**. Апликација је намењена ученицима основне школе и за 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. Кликом на линк **Сазнај више**, потребно је приказати додатне информације о групи инструмената.

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.

# ОСНОВНА ШКОЛА "СОЊА МАРИНКОВИЋ"

Почетна

О аутору

Упутство

## Звуци инструмената

М К  
У У  
З Л  
И Т  
Ч У  
К Р  
А А

Назив инструмента	Слика инструмента	Звук инструмента
Гитара		 Сазнај више
Виолина		 Сазнај више
Клавир		 Сазнај више
Бубњеви		 Сазнај више
Хармоника		 Сазнај више
Контрафагот		 Сазнај више

Copyright © Основна школа "Соња Маринковић"

РАДНИ

# ОСНОВНА ШКОЛА "СОЊА МАРИНКОВИЋ"

Почетна

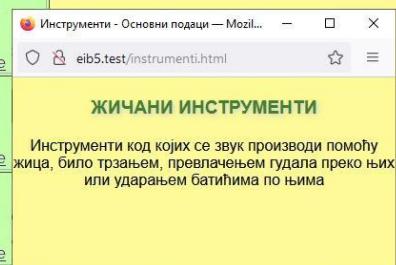
О аутору

Упутство

## Звуци инструмената

М К  
У У  
З Л  
И Т  
Ч У  
К Р  
А А

Назив инструмента	Слика инструмента	Звук инструмента
Гитара		Сазнај више
Виолина		Сазнај више
Клавир		Сазнај више
Бубњеви		Сазнај више
Хармоника		Сазнај више
Контрафагот		Сазнај више



Copyright © Основна школа "Соња Маринковић"

РАДНИ

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б6

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Успех ученика

У склопу припрема за нову школску годину, основна школа „Соња Маринковић“, исказала је потребу за креирањем веб странице са темом приказа успеха ученика за сваки разред.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б6**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

База података **Skola** са табелом **Uspех** која је попуњена подацима о успеху ученика снимљена је на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б6\**. Потребни ресурси.

## ПРИЛОГ:

Креирати веб апликацију са називом **Uspeh učenika** која се састоји од три странице: **Početna**, **Grafika** и **O autoru**.

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 врлодобрих ученика – целобројни без децимала)
- **Dobar** (број добрих ученика – целобројни без децимала)
- **Dovoljan** (број довољних ученика – целобројни без децимала)
- **Nedovoljan** (број недовољних ученика – целобројни без децимала)
- **ProsOcena** (просечна оцена успехај разреда – децимални број са две децимале)

На страници **Početna** табеларно приказати податке о успеху ученика сортиране по просечној оцени коришћењем **GridView** контроле.

На страници **Grafika** 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**.

Трећа страница **O autoru** садржи основне податке о аутору и кратко корисничко упутство и објашњење структуре апликације

CSS код издвојити у посебну екстерну датотеку а учитати је при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

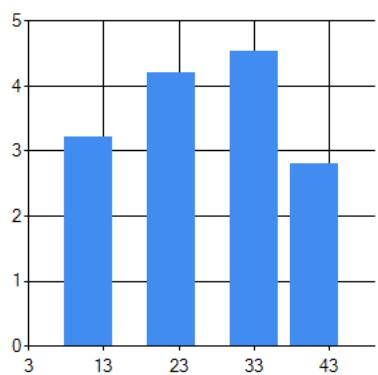
Razred	Odlican	Vrlodobar	Dobar	Dovoljan	Nedovoljan	ProsOcena
11	4	10	3	2	0	3,22
22	6	7	3	4	0	4,21
33	2	4	6	4	0	4,53
41	12	4	4	4	4	2,80

## USPEH UČENIKA

Početna

Grafika

O autoru



© Osnovna škola "Sonja Marinković"

PAUHA BEI

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б7

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Библиотека

Градска библиотека је исказала потребу за креирањем додатка на свој веб сајт, који ће им омогућити брз увид у евиденцију колико је која књига била на читању. Додатак треба да буде креиран у виду веб апликације, која садржи четири странице.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б7**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Датотека biblioteka.xml попуњена подацима снимљена је на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б7\ Потребни ресурси**.

### ПРИЛОГ:

Креирати веб апликацију са називом **Biblioteka** која се састоји од четири странице: **Početna**, **Pregled**, **O autoru** и **Upuststvo**.

Подаци о функционисању библиотеке налазе се у XML фајлу **biblioteka.xml** у следећем облику:

```
<?xml version="1.0" encoding="utf-8" ?>
<biblioteka>
 <knjiga ISBN="10-000000-001">
 <naslov>Baze podataka</naslov>
 <stanje>12</stanje>
 <citano>21</citano>
 </knjiga>
 <knjiga ISBN="10-000000-002">
 <naslov>Programiranje C#</naslov>
 <stanje>24</stanje>
 <citano>19</citano>
 </knjiga>
 <knjiga ISBN="11-000000-003">
 <naslov>Web dizajn 2</naslov>
 <stanje>5</stanje>
 <citano>11</citano>
 </knjiga>
 <knjiga ISBN="11-000000-004">
 <naslov>Web dizajn 1</naslov>
 <stanje>5</stanje>
 <citano>31</citano>
 </knjiga>
 <knjiga ISBN="11-000000-005">
 <naslov>Web programiranje 2</naslov>
 <stanje>5</stanje>
 <citano>51</citano>
 </knjiga>
 <knjiga ISBN="11-000000-006">
 <naslov>Web programiranje 1</naslov>
 <stanje>5</stanje>
 <citano>31</citano>
 </knjiga>
</biblioteka>
```

На страници **Početna** 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 **Pregled** табеларно приказати податке из фајла **biblioteka.xml** сортиране према најчитанијој књизи.

Трећа страница (**O autoru**) садржи основне податке о аутору.

На страници **Upuststvo** написати кратко упутство за коришћење веб апликације.

CSS код издвојити у посебну екстерну датотеку а учитати је при стартовању апликације.

Готову апликацију поставити на веб сервер инсталiran на локалном рачунару.

## BIBLIOTEKA

Početna | Pregled | O autoru | Uputstvo

Možete se logovati sa sledećim nalogima

Korisničko ime: Test  
Lozinka: test123

Korisničko ime: Test2  
Lozinka: test2123

Unesite paremetre za logovanje [Registrujte se](#) ako nemate nalog

Forma za logovanje

Korisničko ime:

Lozinka:

Prijava

© Gradska biblioteka

## BIBLIOTEKA

Početna | Pregled | O autoru | Uputstvo

ISBN	naslov	stanje	citano
11-000000-005	Web programiranje 2	5	51
11-000000-004	Web dizajn 1	5	31
11-000000-006	Web programiranje 1	5	31
10-000000-001	Baze podataka	12	21
10-000000-002	Programiranje C#	24	19
11-000000-003	Web dizajn 2	5	11

© Gradska biblioteka

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б8

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Књига утисака

Музеј Србије је једна савремена установа и прилагођава се захтевима времена у којем живимо и потребама разноврсне публике. У циљу истраживања мишљења посетиоца одлучили су да креирају веб страницу „Књига утисака“, која ће посетиоцима омогућити да електронским путем скажу своје мишљење и сугестије.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б8**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

База података са табелом „Utisak“ снимљена је на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б8\ Потребни ресурси**.

#### ПРИЛОГ:

Креирати веб апликацију са називом ***Knjiga utisaka*** која се састоји од три странице: **Početna, O autoru и Uputstvo**.

На страници **Početna** поставити форму за унос података која треба да има изглед који ја дат на доњој слици.

На страници **O autoru** написати основне податке о аутору.

Трећа страница (**Upustvo**) садржи кратко корисничко упутство и објашњење структуре 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 **Dodaj komentar**, 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.

The screenshot shows a web application interface. At the top, a blue header bar contains the title "KNJIGA UTISAKA". Below the header is a dark navigation bar with three items: "Početna", "O autoru", and "Upustvo". The main content area contains a form for adding a comment. The form has three input fields: "IME:" (Name) with a text input field, "EMAIL:" (Email) with a text input field, and "KOMENTAR:" (Comment) with a large text area. Below the text area is a "Dodaj komentar" (Add comment) button. At the bottom of the page, a grey footer bar contains the copyright notice "© Muzej Srbije".

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б9

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: ЦД каталог

Клуб “Колекционар”, који се бави скупљањем старих албума, исказао је потребу за креирањем веб странице која ћељубитељима музике омогућити претрагу њихове колекције албума по различитим критеријумима.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б9**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Попуњена текстуална датотека која представља колекцију података о албумима, фолдер са слика омота албума и архива слика и икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ\4ЕИТ\_Б9\** Потребни ресурси.

## ПРИЛОГ:

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 **CD katalog** која се састоји од две странице: **Katalog** и **Korisničko uputstvo**.

На страници **Katalog** поставити поља за унос параметара за претраживање података о албумима, тастер **Traži** и линк за прелаз на страницу за упутство. Потребно је омогућити претраживање каталога по следећим пољима:

**Izvođač** – поље за унос текста са одговарајућом ознаком (**Label**),  
**Naziv albuma** - поље за унос текста са одговарајућом ознаком (**Label**),  
**Žanr** - падајућа листа (**DropDownList**) са одговарајућом ознаком (**Label**) са предефинисаним вредностима  
**Godina izdanja** - падајућа листа (**DropDownList**) са одговарајућом ознаком (**Label**) са предефинисаним вредностима  
**Izdavačka kuća** - 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 **Izvođač** узорак „Зор“ апликација треба да излиста податке који у пољу **Izvođač** 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 **Korisničko uputstvo** написати кратко упутство за коришћење веб апликације и линк за прелаз на страницу **Katalog**.

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.

## CD KATALOG

Katalog Korisničko uputstvo

Izvođač:

Naziv albuma:

Žanr:

Godina izdanja:

Izdavačka kuća:

Izvođač	Naziv albuma	Žanr	Godina izdavanja	Izdavačka kuća	Slika omota
Vrisak generacije	Ponekad mi dodje da puknem od jada	Punk	2007	SKC Novi Sad	
Dva minuta mrznje	Moderna omladina	Punk	2008	SKC Novi Sad	
Atheist Rap	20 godina stihije - ujutru smo mamurni i gadni	Punk-Rock	2009	SKC Novi Sad	
Blitzkrieg	2nd	Punk	2009	SKC Novi Sad	
Goblini	Crno na belo	Punk	2010	SKC Novi Sad	
Ritam nereda	Paralelni svet	Rock	2010	SKC Novi Sad	

Traži

© Klub Kolekcionar

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б10

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Електронски речник

У склопу припрема за нову школску годину, основна школа „Соња Маринковић“, исказала је потребу за креирањем веб странице за праћење наставе из енглеског језика у виду електронског речника, који ће наставницима енглеског језика олакшати обраду нових непознатих речи.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
  - Обезбедити функционалности веб апликације;
  - Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
  - Написати упутство корисницима о коришћењу апликације;
  - Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б10**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

База података са предефинисаним скупом података снимљена је на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б10\ Потребни ресурси**.

РАДНА ВРСАЈА

## ПРИЛОГ:

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 **Elektronski rečnik** која се састоји од три странице: **Rečnik**, **Dodavanje novih reči** и **Uputstvo**.

На страници **Rečnik** поставити поља за избор смера превођења (Српско – Енглески, Енглеско - Српски), поље за унос речи за превођење, поља за приказ превода и описа, тастер **Prevedi** 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 **Prevedi** приказати одговарајући превод унете речи и опис уколико постоје у бази података.

На страници за **Dodavanje novih reči** у речник поставити поља за унос енглеске речи, одговарајуће српске речи, поље за унос описа, тастер **Snimi** и линк за прелазак на страницу **Rečnik**. Притиском на тастер **Snimi** подаци се снимају у базу података.

На страници **Uputstvo** написати кратко упутство за коришћење веб апликације.

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.

The screenshot shows the user interface of the 'Elektronski Rečnik' application. At the top, there is a blue header bar with the title 'ELEKTRONSKI REČNIK'. Below the header, there is a dark blue navigation bar containing three links: 'Rečnik', 'Dodavanje novih reči', and 'Uputstvo'. The main content area has a light gray background. It contains several input fields and a button. On the left, there is a dropdown menu labeled 'Smer:' with the placeholder 'Selektuj smer prevođenja'. Below it are two text input fields: 'Engleska reč:' and 'Srpska reč:', each with a corresponding empty input box. To the right of these is a large, empty rectangular text area. At the bottom left of this area, there is a label 'Opis:' followed by a text input field with a small '...' icon. At the bottom right of the main content area, there is a blue button labeled 'Prevedi'. At the very bottom of the page, there is a footer bar with the copyright notice '© Osnovna škola "Sonja Marinković"'.

## ELEKTRONSKI REČNIK

Rečnik

Dodavanje novih reči

Uputstvo

Engleska reč:

Srpska reč:

  
Opis: 

© Osnovna škola "Sonja Marinković"

PAUHA BEI

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б11

НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Временска прогноза

Туристичка агенција “Планинар”, исказала је потребу за креирањем веб странице са темом Временска прогноза.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б11**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Архива слика и икона снимљени су на радну површину рачунара (desktop), у фолдеру **Матурски\_испит\_4ЕИТ \4ЕИТ\_Б11\ Потребни ресурси**.

## ПРИЛОГ:

Креирати веб апликацију са називом **Vremenska prognoza** која се састоји од три странице: **Početna**, **O autoru** и **Uputstvo**.

На страници **Početna** 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. Након избора града, поред временске прогнозе, поставити и кратак опис најзначајнијих туристичких локација, које би требало посетити у том граду односно дестинацији.

Веб апликација треба да има следећу функционалност: На форми омогућити избор најмање за десет градова или локација у Србији. По избору локације или града треба 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.

На страници **O autoru** написати основне податке о аутору.

Трећа страница **Uputstvo** садржи кратко корисничко упутство и објашњење структуре апликације.

## Turistička agencija "Planinar"

Početna

O autoru

Uputstvo

Izaberite grad

**naslovi.net**  Pretraži

Politika Ekonomija Svet Balkan Društvo Hronika Kultura

Beta N1 B92 BBC News Blic Nova Politika RTS Danas Novosti

### Vremenska prognoza

**Beograd 18°C**  
**Pretežno oblačno**

Vlažnost: 83%  
Padavine: 0.1 mm  
Vetar: 2 m/s  
Pritisak: 994 mb  
Izmereno u 01:10

**Vreme u naredna 24h**

03h	06h	09h	12h	15h	18h	21h
16°	14°	17°	22°	22°	19°	1°

© Turistička agencija "Planinar"

## Turistička agencija "Planinar"

Početna

O autoru

Uputstvo

The screenshot shows a weather forecast for Novi Sad. At the top, there is a search bar labeled "Izaberite grad" with "Novi Sad" selected. Below it, the "naslovinet" logo is visible, along with a navigation menu for "Politika", "Ekonomija", "Svet", "Balkan", "Društvo", and "Hronika". A sub-menu for "Beta", "N1", "B92", "BBC News", "Blic", "Nova", "Politika", "RTS", and "Dana" is also present. The main content area features a section titled "Vremenska prognoza" with a yellow crescent moon icon. It displays the current weather in Novi Sad: "Novi Sad 14°C" and "Vedro". Below this, detailed weather information is provided: "Vlažnost: 82%", "Padavine: -", "Vetar: 2 m/s", and "Pritisak: 1013 mb". It also notes that the data was "Izmereno u 00:10". A second section titled "Vreme u naredna 24h" shows a 24-hour forecast with icons and temperatures: 03h (14°), 06h (15°), 09h (22°), 12h (29°), 15h (26°), and 18h (2°). To the right of the main content, a separate window titled "Planinar | Obilazak — Mo..." shows a map of Novi Sad with the text "Novi Sad" and a list of attractions: "Katedrala, mostovi, Dunavska ulica, Strand, Petrovaradinska tvrđava...".

© Turistička agencija "Planinar"

РАДНИК

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б12

НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Резервација аутобуских карата

У циљу побољшања својих услуга грађанима, Аутобуска станица „Нови Сад”, исказала је потребу за креирањем веб странице за резервацију аутобуских карата.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б12**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

РАДНА ВРСТАДИЈА

## ПРИЛОГ:

Дата је база података која има једну табелу у којој се памти резервација седишта у аутобусу. Табела има следећу структуру:

```
[ID] [int] IDENTITY(1,1) NOT NULL,
[BrojSedista] [int] NOT NULL,
[Rezervacija] [bit] NOT NULL,
```

Креирати веб апликацију са називом **Rezervacija autobuskih karata** која се састоји од три странице: **Početna**, **O autoru** и **Uputstvo**.

На страници **Početna** поставити приказ нумерисаних седишта аутобуса почев од броја 2 па до 52, разврстаних у четири колоне. Испод приказа седишта поставити форму за слање мейла која треба да има следећа поља: **Broj sedišta, Ime i prezime, E-mail**.

На страници **O autoru** написати основне податке о аутору.

Трећа страница (**Uputstvo**) садржи кратко корисничко упутство и објашњење структуре 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 **Broj sedišta** аутоматски се уноси податак о седишту које желимо да резервишемо. Када се форма попуни осталим подацима и пошаље, слободно седиште мења боју у црвену, то јест постаје резервисано и у бази се мења поље резервација.

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.

## REZERVACIJA AUTOBUSKIH KARATA

[Početna](#)[O autoru](#)[Uputstvo](#)

PRIKAZ SEDIŠTA AUTOBUSA:

2	3
6	7
10	11
14	15
18	19
22	23
26	27
30	31
34	35
38	39
42	43
46	47
50	51
4	5
8	9
12	13
16	17
20	21
24	25
28	29
32	33
36	37
40	41
44	45
48	49
52	53

Broj sedišta:

Ime i prezime:

E-mail:

© Autobuska stanica

PAHTI

## REZERVACIJA AUTOBUSKIH KARATA

Početna    O autoru    Uputstvo

PRIKAZ SEDIŠTA AUTOBUSA:

2	3
6	7
10	11
14	15
18	19
22	23
26	27
30	31
34	35
38	39
42	43
46	47
50	51
4	5
8	9
12	13
16	17
20	21
24	25
28	29
32	33
36	37
40	41
44	45
48	49
52	53

Broj sedišta:

Ime i prezime:

E-mail:

© Autobuska stanica

PAUHNIK

**ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б13**

**НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Телефонски именик**

Агенција за продају некретнина исказала је потребу за креирањем веб апликације за брзу претрагу контакт података о корисницима.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б13**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Текстуални фајл са подацима о корисницима дат је као **Прилог\_4ЕИТ\_Б131**

## ПРИЛОГ:

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 именника, тастер **Тражи** и линкове за прелаз на остале странице. Потребно је омогућити претраживање именика по следећим пољима:

<b>Име</b>	– поље за унос текста са одговарајућом лабелом,
<b>Презиме</b>	– поље за унос текста са одговарајућом лабелом,
<b>Адреса</b>	– поље за унос текста са одговарајућом лабелом,
<b>Место</b>	– падајућа листа са одговарајућом лабелом ( <b>DropDownList</b> ) са префинисаним вредностима
<b>Број телефона</b>	– 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 инсталиран на локалном рачунару.

Parametri za pretragu:

Ime: <input type="text"/>	Adresa: <input type="text"/>	Telefon: <input type="text"/>	<input type="button" value="Traži"/>
Prezime: <input type="text"/>	Mesto: <input type="text"/>	Email: <input type="text"/>	

Rbr	Ime	Prezime	Adresa	Telefon	E-mail	Mesto
1.	Pera	Peric	Neznanog junaka bb	018 123- 456	pera@email.com	Niš
2.	Mika	Mikic	Visegradska bb	018 987- 475	mika@email.com	Niš
3.	Srdjan	Simic	Sveznalacka bb	018 954- 475	srdjan@email.com	Niš
4.	Laza	Lazarevic	Beogradska bb	018 954- 225	laza@email.com	Beograd

Elektrotehnička škola "Nikola Tesla" Niš

[Važni telefoni](#) [Uputstvo](#) [O autoru](#)

### Uputstvo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec faucibus lacinia ipsum, et sollicitudin tortor pulvinar non. Aenean rutrum molestie ipsum, id tristique eros placerat vel. Nulla dui nunc, fringilla vel enim eget, gravida placerat nibh. Proin sem massa, pharetra scelerisque posuere sed, fringilla non risus. Nunc consectetur sagittis elit, ac blandit diam bibendum non. Integer consectetur volutpat fermentum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec finibus metus metus, at porta ligula dictum non. Vestibulum ut quam dictum, finibus nibh et, posuere tortor. Quisque fringilla urna eu interdum tristique. Sed sed enim nulla. Donec semper sit amet est sit amet maximus.

Praesent pharetra sit amet massa id pellentesque. Nulla luctus urna nisi, ac elementum ante commodo at. Nulla eu est fringilla mi laoreet luctus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam purus lorem, blandit ut sem id, fermentum porttitor arcu. Nullam eget pulvinar elit. Cras elementum facilisis tellus eu consequat. Duis facilisis id orci sit amet iaculis. Phasellus mattis sodales justo id cursus. Vestibulum nec vestibulum libero. Nunc dolor lorem, dictum in lacus nec, venenatis euismod purus. Duis non facilisis sapien, non faucibus diam. Curabitur lacinia lacinia tempor. Curabitur ullamcorper sapien ac ante viverra, nec placerat velit accumsan. Aliquam quis nulla vel tellus luctus pellentesque vitae eleifend mauris. Etiam pharetra, lectus id efficitur rhoncus, odio nulla ultrices justo, vel consectetur nulla erat non nulla.

Maecenas justo dui, maximus eget dignissim ac, fermentum a lacus. Cras quis condimentum lectus, quis molestie ex. Fusce sagittis pulvinar odio, vel pretium arcu ultrices non. Sed justo purus, semper vel tincidunt eu, commodo a orci. Quisque nec porta lacus. Duis sit amet nulla imperdiet, rhoncus felis nec, euismod diam. Sed porttitor blandit aliquet. Proin ut risus vitae diam tempor rutrum.

Elektrotehnička škola "Nikola Tesla" Niš

[Početna](#) [O autoru](#)

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б14

НАЗИВ РАДНОГ ЗАДАТКА: **Израда веб апликације: ТВ програм**

За потребе ТВ станице креирати веб апликацију за приказивање дневне програмске шеме.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б14**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Архива са текстуалним фајловима који садрже програмске шеме по данима дата је као **Прилог\_4ЕИТ\_Б141**.

Архива слика и икона дата је као **Прилог\_4ЕИТ\_Б00**.

## ПРИЛОГ:

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 | Фilm | C:\TV Program\Slake\Film.png  
17:00 | Вести | Информативни програм | C:\TV Program\Slake\Informativni.png  
17:15 | Тесна кожа 2 | Фilm | 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, 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 инсталiran на локалном рачунару.

Zadatak 14 | TV Program

Datum: 20.09.2021 | Tip emisije: | Pronadji

Vreme	Naziv Emisije	Tip Emisije
06:20	Zvala se Feriha	
07:00	Ovčica Šone	
07:45	EXploziv	
08:05	Moja mama kuva bolje od tvoje	
10:15	Neki novi klinci	
12:00	Praktična žena	
13:00	Vesti	
13:15	Andrija i Anđelka specijal	
14:00	Mehaničar	

Elektrotehnička škola "Nikola Tesla" Niš | [Uputstvo](#) | [O autoru](#)

### Uputstvo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec faucibus lacinia ipsum, et sollicitudin tortor pulvinar non. Aenean rutrum molestie ipsum, id tristique eros placerat vel. Nulla dui nunc, fringilla vel enim eget, gravida placerat nibh. Proin sem massa, pharetra scelerisque posuere sed, fringilla non risus. Nunc consectetur sagittis elit, ac blandit diam bibendum non. Integer consectetur volutpat fermentum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec finibus metus metus, at porta ligula dictum non. Vestibulum ut quam dictum, finibus nibh et, posuere tortor. Quisque fringilla urna eu interdum tristique. Sed sed enim nulla. Donec semper sit amet est sit amet maximus.

Praesent pharetra sit amet massa id pellentesque. Nulla luctus urna nisi, ac elementum ante commodo at. Nulla eu est fringilla mi laoreet luctus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam purus lorem, blandit ut sem id, fermentum porttitor arcu. Nullam eget pulvinar elit. Cras elementum facilisis tellus eu consequat. Duis facilisis id orci sit amet iaculis. Phasellus mattis sodales justo id cursus. Vestibulum nec vestibulum libero. Nunc dolor lorem, dictum in lacus nec, venenatis euismod purus. Duis non facilisis sapien, non faucibus diam. Curabitur lacinia lacinia tempor. Curabitur ullamcorper sapien ac ante viverra, nec placerat velit accumsan. Aliquam quis nulla vel tellus luctus pellentesque vitae eleifend mauris. Etiam pharetra, lectus id efficitur rhoncus, odio nulla ultrices justo, vel consectetur nulla erat non nulla.

Maecenas justo dui, maximus eget dignissim ac, fermentum a lacus. Cras quis condimentum lectus, quis molestie ex. Fusce sagittis pulvinar odio, vel pretium arcu ultrices non. Sed justo purus, semper vel tincidunt eu, commodo a orci. Quisque nec porta lacus. Duis sit amet nulla imperdiet, rhoncus felis nec, euismod diam. Sed porttitor blandit aliquet. Proin ut risus vitae diam tempor rutrum.

РАДНА РАБОТА

## ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б15

### НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Ред вожње

За потребе предузећа за превоз путника потребно је развити веб апликацију за праћење реда вожње аутобуса.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б15**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 150 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Текстуални фајлови са списовима полазака дати су као **Прилог\_4ЕИТ\_Б151**

## ПРИЛОГ:

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, назлова 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.

Linija:  Smer:

Polasci:

**Rbr Polazak**

1. 06:00
2. 06:15
3. 06:20

Elektrotehnička škola "Nikola Tesla" Niš

[Kontakt](#) [Uputstvo](#) [O autoru](#)

### Uputstvo

Lore ipsum dolor sit amet, consectetur adipiscing elit. Donec faucibus lacinia ipsum, et sollicitudin tortor pulvinar non. Aenean rutrum molestie ipsum, id tristique eros placerat vel. Nulla dui nunc, fringilla vel enim eget, gravida placerat nibh. Proin sem massa, pharetra scelerisque posuere sed, fringilla non risus. Nunc consectetur sagittis elit, ac blandit diam bibendum non. Integer consectetur volutpat fermentum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec finibus metus metus, at porta ligula dictum non. Vestibulum ut quam dictum, finibus nibh et, posuere tortor. Quisque fringilla urna eu interdum tristique. Sed sed enim nulla. Donec semper sit amet est sit amet maximus.

Praesent pharetra sit amet massa id pellentesque. Nulla luctus urna nisi, ac elementum ante commodo at. Nulla eu est fringilla mi laoreet luctus. Lore ipsum dolor sit amet, consectetur adipiscing elit. Etiam purus lorem, blandit ut sem id, fermentum porttitor arcu. Nullam eget pulvinar elit. Cras elementum facilisis tellus eu consequat. Duis facilisis id orci sit amet iaculis. Phasellus mattis sodales justo id cursus. Vestibulum nec vestibulum libero. Nunc dolor lorem, dictum in lacinia nec, venenatis euismod purus. Duis non facilisis sapien, non faucibus diam. Curabitur lacinia lacinia tempor. Curabitur ullamcorper sapien ac ante viverra, nec placerat velit accumsan. Aliquam quis nulla vel tellus luctus pellentesque vitae eleifend mauris. Etiam pharetra, lectus id efficitur rhoncus, odio nulla ultrices justo, vel consectetur nulla erat non nulla.

Maecenas justo dui, maximus eget dignissim ac, fermentum a lacinia. Cras quis condimentum lectus, quis molestie ex. Fusce sagittis pulvinar odio, vel pretium arcu ultrices non. Sed justo purus, semper vel tincidunt eu, commodo a orci. Quisque nec porta lacinia. Duis sit amet nulla imperdiet, rhoncus felis nec, euismod diam. Sed porttitor blandit aliquet. Proin ut risus vitae diam tempor rutrum.

Elektrotehnička škola "Nikola Tesla" Niš

[Početna](#) [O autoru](#)

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б16

НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Распоред часова

У склопу припрема за нову школску годину, основна школа „Бубањски хероји“, исказала је потребу за креирањем веб апликације за приказ актуелног распореда часова.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б16**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Текстуални фајл са рапоредом часова дат је као **Прилог\_4ЕИТ\_Б161**

## ПРИЛОГ:

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 инсталиран на локалном рачунару.

Raspored Učitaj XML					
Rbr	Ponedeljak	Utorak	Sreda	Četvrtak	Petak
1	Tehnicka dokumentacija	Engleski jezik	Programiranje	Elektronsko poslovanje	Internet tehnologije i servisi
2	Tehnicka dokumentacija	Engleski jezik	Programiranje	Elektronsko poslovanje	Internet tehnologije i servisi
3	Zastita informacionih Sistema	Zastita informacionih sistema	Programiranje	Filozofija	Fizicko
4	Elektronsko poslovanje	Zastita informacionih sistema	Web programiranje	Izborna matematika	Srpski jezik
5	Preduzetnistvo	Fizicko	Web programiranje	Srpski jezik	Veronauka/Gradjansko
6	Preduzetnistvo	Matematika	Web programiranje	Srpski jezik	/
7	Internet tehnologije i servisi	Matematika	/	Matematika	/

## Raspored časova

Rbr	Ponedeljak	Utorak	Sreda	Četvrtak	Petak
1	Tehnicka dokumentacija	Engleski jezik	Programiranje	Elektronsko poslovanje	Internet tehnologije i servisi
2	Tehnicka dokumentacija	Engleski jezik	Programiranje	Elektronsko poslovanje	Internet tehnologije i servisi
3	Zastita informacionih Sistema	Zastita informacionih sistema	Programiranje	Filozofija	Fizicko
4	Elektronsko poslovanje	Zastita informacionih sistema	Web programiranje	Izborna matematika	Srpski jezik
5	Preduzetnistvo	Fizicko	Web programiranje	Srpski jezik	Veronauka/Gradjansko
6	Preduzetnistvo	Matematika	Web programiranje	Srpski jezik	/
7	Internet tehnologije i servisi	Matematika	/	Matematika	/

© 2021 - ETŠ "Nikola Tesla" Niš

Učitaj XML fajl

XML datoteka

© 2021 - ETŠ "Nikola Tesla" Niš

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б17

НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Веб продавница

Продавница за продају мобилних телефона изказала је потребу за креирањем веб апликације за приказ и претрагу података о аркилима у свом асортиману.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б17**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Текстуални фајл са подацима о уређајима дат је као **Прилог\_4ЕИТ\_Б171**

Архива слика дата је као **Прилог\_4ЕИТ\_Б00**.

## ПРИЛОГ:

Дат је фолдер са сликама производа и текстуална датотека под називом **vebprodavnica.txt** која представља колекцију података о артиклима веб проавнице.

Сваки ред у текстуалној датотеци представља податке о једном артиклу у облику:

- **Шифра артикла** – памти се као алфанимерички податак у сваком реду текстуалне датотеке на позицији од индекса 0 до 5
- **Назив артикла** - памти се као алфанимерички податак у сваком реду текстуалне датотеке на позицији од индекса 6 до 30
- **Произвођач** - памти се као алфанимерички податак у сваком реду текстуалне датотеке на позицији од индекса 31 до 50
- **RAM меморија** - памти се као алфанимерички податак у сваком реду текстуалне датотеке на позицији од индекса 51 до 55
- **Тип процесора** - памти се као алфанимерички податак у сваком реду текстуалне датотеке на позицији од индекса 56 до 70
- **Камера** - памти се као алфанимерички 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 реализацији као падајуће листе са одговарајућом лабелом (**DropDownList**) и предефинисаним вредностима (нпр. поље за избор производа напунити свим различитим производима 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 инсталиран на локалном рачунару.



## Web prodavnica

Parametri za pretragu:

Proizvođač:  RAM:

Kamera:  Dual SIM:

Ekran:  Procesor:

Traži

	Karakteristike	Cena
	<p><b>ALCATEL ONE TOUCH 5020X-M POP</b></p> <p>Proizvođač: Alcatel RAM: 512 Procesor: Single Core Ekran: 4 Kamera: 5 Dual SIM: NE</p>	<b>11.990 RSD</b>
	<p><b>Lenovo K900, 32GB, Steel Gray</b></p> <p>Proizvođač: Lenovo RAM: 2 Procesor: Dual Core Ekran: 5.5 Kamera: 13 Dual SIM: NE</p>	<b>49.990 RSD</b>

PAPIL'

## Uputstvo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec faucibus lacinia ipsum, et sollicitudin tortor pulvinar non. Aenean rutrum molestie ipsum, id tristique eros placerat vel. Nulla dui nunc, fringilla vel enim eget, gravida placerat nibh. Proin sem massa, pharetra scelerisque posuere sed, fringilla non risus. Nunc consectetur sagittis elit, ac blandit diam bibendum non. Integer consectetur volutpat fermentum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec finibus metus metus, at porta ligula dictum non. Vestibulum ut quam dictum, finibus nibh et, posuere tortor. Quisque fringilla urna eu interdum tristique. Sed sed enim nulla. Donec semper sit amet est sit amet maximus.

Praesent pharetra sit amet massa id pellentesque. Nulla luctus urna nisi, ac elementum ante commodo at. Nulla eu est fringilla mi laoreet luctus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam purus lorem, blandit ut sem id, fermentum porttitor arcu. Nullam eget pulvinar elit. Cras elementum facilisis tellus eu consequat. Duis facilisis id orci sit amet iaculis. Phasellus mattis sodales justo id cursus. Vestibulum nec vestibulum libero. Nunc dolor lorem, dictum in lacus nec, venenatis euismod purus. Duis non facilisis sapien, non faucibus diam. Curabitur lacinia lacinia tempor. Curabitur ullamcorper sapien ac ante viverra, nec placerat velit accumsan. Aliquam quis nulla vel tellus luctus pellentesque vitae eleifend mauris. Etiam pharetra, lectus id efficitur rhoncus, odio nulla ultrices justo, vel consectetur nulla erat non nulla.

Maecenas justo dui, maximus eget dignissim ac, fermentum a lacus. Cras quis condimentum lectus, quis molestie ex. Fusce sagittis pulvinar odio, vel pretium arcu ultrices non. Sed justo purus, semper vel tincidunt eu, commodo a orci. Quisque nec porta lacus. Duis sit amet nulla imperdiet, rhoncus felis nec, euismod diam. Sed porttitor blandit aliquet. Proin ut risus vitae diam tempor rutrum.

РАДНА РАБОТА

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б18

НАЗИВ РАДНОГ ЗАДАТКА: **Израда веб апликације: Временска прогноза**

На портал новинске агенције потребно је додати функционалност приказивања података о временској прогнози по местима.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б18**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Текстуални фајл са списком података о временској прогнози дат је као **Прилог\_4ЕИТ\_Б181**

Архива слика дата је као **Прилог\_4ЕИТ\_Б00**.

## ПРИЛОГ:

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 инсталiran на локалном рачунару.

Sifra	NazivMesta	MaxTemperatura
001	Nis	10
002	Beograd	7
003	Kopaonik	0
004	Vranje	10
005	Kraljevo	17
006	Novi Sad	2
007	Loznica	20
008	Bor	17
009	Subotica	15
010	Krusevac	20
011	Leskovac	23
012	Aleksinac	13

© 2021 - ETŠ "Nikola Tesla" Niš

Maksimalne temperature Vremenska prognoza

### Prognoza

Mesto

Šifra mesta	002
Naziv mesta	Beograd
Min temperatura	1
Max temperatura	7



© 2021 - ETŠ "Nikola Tesla" Niš

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б19

НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Конвертор текста

Онлајн портал за превођење и пресловљавање текста исказао је потребу за проширењем функционалности постојеће веб апликације креирањем веб сервиса за превођење текста из ћирилице у латиницу и обратно.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б19**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

РАДНА ВЕРСИЈА

## ПРИЛОГ:

Креирати веб сервис који има две јавне веб методе:

- *string CirilicaULatinicu(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 инсталiran на локалном рачунару.

Zadatak 9 Konvertor teksta

Tekst na latinici:  
Ovo je tekst koji treba prevesti na cirilicu.

Prevedu na cirilicu

Tekst na cirilici:  
Ово је текст који треба превести на ћирилицу.

Prevedu na latinicu

Elektrotehnička škola "Nikola Tesla" Niš [Uputstvo](#) [O autoru](#)

## Uputstvo

Lore ipsum dolor sit amet, consectetur adipiscing elit. Donec faucibus lacinia ipsum, et sollicitudin tortor pulvinar non. Aenean rutrum molestie ipsum, id tristique eros placerat vel. Nulla dui nunc, fringilla vel enim eget, gravida placerat nibh. Proin sem massa, pharetra scelerisque posuere sed, fringilla non risus. Nunc consectetur sagittis elit, ac blandit diam bibendum non. Integer consectetur volutpat fermentum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec finibus metus metus, at porta ligula dictum non. Vestibulum ut quam dictum, finibus nibh et, posuere tortor. Quisque fringilla urna eu interdum tristique. Sed sed enim nulla. Donec semper sit amet est sit amet maximus.

Praesent pharetra sit amet massa id pellentesque. Nulla luctus urna nisi, ac elementum ante commodo at. Nulla eu est fringilla mi laoreet luctus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam purus lorem, blandit ut sem id, fermentum porttitor arcu. Nullam eget pulvinar elit. Cras elementum facilisis tellus eu consequat. Duis facilisis id orci sit amet iaculis. Phasellus mattis sodales justo id cursus. Vestibulum nec vestibulum libero. Nunc dolor lorem, dictum in lacus nec, venenatis euismod purus. Duis non facilisis sapien, non faucibus diam. Curabitur lacinia lacinia tempor. Curabitur ullamcorper sapien ac ante viverra, nec placerat velit accumsan. Aliquam quis nulla vel tellus luctus pellentesque vitae eleifend mauris. Etiam pharetra, lectus id efficitur rhoncus, odio nulla ultrices justo, vel consectetur nulla erat non nulla.

Maecenas justo dui, maximus eget dignissim ac, fermentum a lacus. Cras quis condimentum lectus, quis molestie ex. Fusce sagittis pulvinar odio, vel pretium arcu ultrices non. Sed justo purus, semper vel tincidunt eu, commodo a orci. Quisque nec porta lacus. Duis sit amet nulla imperdiet, rhoncus felis nec, euismod diam. Sed porttitor blandit aliquet. Proin ut risus vitae diam tempor rutrum.

РАДНА РАБОТА

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б20

НАЗИВ РАДНОГ ЗАДАТКА: **Израда веб апликације: Курсна листа**

На постојећем сајту мењачнице „Срећни динар“ потребно је додати нову услугу приказивања курсне листе на жељени дан.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б20**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **4ЕИТ** на десктопу рачунара.

Текстуални фајл са подацима о курсу валута дат је као **Прилог\_4ЕИТ\_Б201**

## ПРИЛОГ:

Написати 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:

**Datum | Oznaka valute | Kurs**

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 подatak.

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 инсталиран на локалном рачунару.

Zadatak 10

Kursna lista

Datum:

Valuta:

**Kurs na dan: 117.58**

Elektrotehnička škola "Nikola Tesla" Niš

[Unos kursa](#)

[O autoru](#)

Zadatak 10

Kursna lista

### Unos kursa

Datum:

Valuta:

Iznos:

Elektrotehnička škola "Nikola Tesla" Niš

[Kurs na dan](#)

[O autoru](#)

PAIIHA

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б21

НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Сортирање табеле

За потребе Републичког завода за статистику креирати веб сајт који ће имати могућност табеларног приказа и сортирања података.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б21**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **Прилог\_4ЕИТ\_Б21** на десктопу рачунара.

РАДНА ВРСАДА

## ПРИЛОГ:

Дата је *HTML* табела са три колоне (*Рбр, Име, Презиме*) на веб страници са следећим подацима:

```
<table id="Tabela">

<tr>

 <th>Rbr</th>
 <th>Ime</th>
 <th>Prezime</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 **O 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.

Zadatak 21 | Sortiranje tabele

## SORTIRANJE TABELE

Rbr	Ime	Prezime
1.	Pera	Mikić
4.	Dragan	Milenković
3.	Žika	Perić
5.	Pavle	Tošić
2.	Mika	Žikić

Elektrotehnička škola "Nikola Tesla" Niš [Uputstvo](#) [O autoru](#)

## Uputstvo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec faucibus lacinia ipsum, et sollicitudin tortor pulvinar non. Aenean rutrum molestie ipsum, id tristique eros placerat vel. Nulla dui nunc, fringilla vel enim eget, gravida placerat nibh. Proin sem massa, pharetra scelerisque posuere sed, fringilla non risus. Nunc consectetur sagittis elit, ac blandit diam bibendum non. Integer consectetur volutpat fermentum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec finibus metus metus, at porta ligula dictum non. Vestibulum ut quam dictum, finibus nibh et, posuere tortor. Quisque fringilla urna eu interdum tristique. Sed sed enim nulla. Donec semper sit amet est sit maximus.

Praesent pharetra sit amet massa id pellentesque. Nulla luctus urna nisi, ac elementum ante commodo at. Nulla eu est fringilla mi laoreet luctus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam purus lorem, blandit ut sem id, fermentum porttitor arcu. Nullam eget pulvinar elit. Cras elementum facilisis tellus eu consequat. Duis facilisis id orci sit amet iaculis. Phasellus mattis sodales justo id cursus. Vestibulum nec vestibulum libero. Nunc dolor lorem, dictum in lacus nec, venenatis euismod purus. Duis non facilisis sapien, non faucibus diam. Curabitur lacinia lacinia tempor. Curabitur ullamcorper sapien ac ante viverra, nec placerat velit accumsan. Aliquam quis nulla vel tellus luctus pellentesque vitae eleifend mauris. Etiam pharetra, lectus id efficitur rhoncus, odio nulla ultrices justo, vel consectetur nulla erat non nulla.

Maecenas justo dui, maximus eget dignissim ac, fermentum a lacus. Cras quis condimentum lectus, quis molestie ex. Fusce sagittis pulvinar odio, vel pretium arcu ultrices non. Sed justo purus, semper vel tincidunt eu, commodo a orci. Quisque nec porta lacus. Duis sit amet nulla imperdiet, rhoncus felis nec, euismod diam. Sed porttitor blandit aliquet. Proin ut risus vitae diam tempor rutrum.

PAIIHA

ШИФРА РАДНОГ ЗАДАТКА: 4ЕИТ-Б22

НАЗИВ РАДНОГ ЗАДАТКА: Израда веб апликације: Ефекти табеле

За потребе анализе резултата такмичења из програмирања креирати веб сајт који ће имати могућност табеларног приказа и манипулације подацима.

На основу спецификације дате у **Прилогу задатка** урадити следеће:

- Дизајнирати веб апликацију у складу са описом захтева;
- Обезбедити функционалности веб апликације;
- Код и ресурсе веб апликације организовати у засебне датотеке и фолдере;
- Написати упутство корисницима о коришћењу апликације;
- Апликацију запамтити на десктопу рачунара, у фолдеру **Матурски\_испит\_4ЕИТ\Име\_Презиме\4ЕИТ\_Б22**, где се као име и презиме уноси име ученика.

Предвиђено време за израду задатка је 120 минута.

По истеку максималног времена задатак се прекида и бодује се оно што је до тада урађено.

У оквиру времена за израду задатка ученик може да одустане од даљег рада, при чему се бодује оно што је до тада урађено.

Прилоге преузети из фолдера **Прилог\_4ЕИТ\_Б221** на десктопу рачунара.

РАДНА ВРСАДА

## ПРИЛОГ:

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 се табела 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 **O апликацији** на којој се налазе информације о аутору и апликацији (име, презиме и имејл ученика, име апликације и датум креирања).

На страници **O 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.

Zadatak 22
Efekti tabele

## EFEKTI TABELE

Rbr	Ime	Prezime	Škola	Poeni
3.	Milan	Markovic	OŠ "Mika Antic"	73
5.	Jelana	Peric	OŠ "Milan Milutinovic"	68
1.	Pera	Mikic	OŠ "Vuk Karadžić"	54
4.	Ivan	Tosic	OŠ "Bubanjski heroji"	21
2.	Mika	Zikic	OŠ "Ivo Andrić"	20

Sortiraj
Oboji neparne
Položili
Nisu položili
Izdvoji one koji su položili

Elektrotehnička škola "Nikola Tesla" Niš
Uputstvo
O autoru

## Uputstvo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec faucibus lacinia ipsum, et sollicitudin tortor pulvinar non. Aenean rutrum molestie ipsum, id tristique eros placerat vel. Nulla dui nunc, fringilla vel enim eget, gravida placerat nibh. Proin sem massa, pharetra scelerisque posuere sed, fringilla non risus. Nunc consectetur sagittis elit, ac blandit diam bibendum non. Integer consectetur volutpat fermentum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec finibus metus metus, at porta ligula dictum non. Vestibulum ut quam dictum, finibus nibh et, posuere tortor. Quisque fringilla urna eu interdum tristique. Sed sed enim nulla. Donec semper sit amet est sit maximus.

Praesent pharetra sit amet massa id pellentesque. Nulla luctus urna nisi, ac elementum ante commodo at. Nulla eu est fringilla mi laoreet luctus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam purus lorem, blandit ut sem id, fermentum porttitor arcu. Nullam eget pulvinar elit. Cras elementum facilisis tellus eu consequat. Duis facilisis id orci sit amet iaculis. Phasellus mattis sodales justo id cursus. Vestibulum nec vestibulum libero. Nunc dolor lorem, dictum in lacus nec, venenatis euismod purus. Duis non facilisis sapien, non faucibus diam. Curabitur lacinia lacinia tempor. Curabitur ullamcorper sapien ac ante viverra, nec placerat velit accumsan. Aliquam quis nulla vel tellus luctus pellentesque vitae eleifend mauris. Etiam pharetra, lectus id efficitur rhoncus, odio nulla ultrices justo, vel consectetur nulla erat non nulla.

Maecenas justo dui, maximus eget dignissim ac, fermentum a lacus. Cras quis condimentum lectus, quis molestie ex. Fusce sagittis pulvinar odio, vel pretium arcu ultrices non. Sed justo purus, semper vel tincidunt eu, commodo a orci. Quisque nec porta lacus. Duis sit amet nulla imperdiet, rhoncus felis nec, euismod diam. Sed porttitor blandit aliquet. Proin ut risus vitae diam tempor rutrum.

PAIIHA

## ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА А

Шифра радног задатка	
Назив радног задатка	
Назив школе	
Седиште	
Образовни профил	
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:								Укупно бодова
Аспекти	1.1	1.2	2.1	2.2	2.3	3.1	3.2	
Бодови								
Аспекти	4.1							
Бодови								

Члан испитне комисије:	Место и датум:
------------------------	----------------

### КОМЕНТАРИ:

## **ЗА СВАКИ ИНДИКАТОР ЗАОКРУЖИТИ ОДГОВАРАЈУЋИ БРОЈ БОДОВА**

### **1. Припрема и организација рада**

#### **1.1. Припремање и одржавање радног места**

<b>ИНДИКАТОРИ</b> (максималан број бодова 6)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Извршен избор и подешавања радног окружења	2	0
Преузети потребни ресурси	2	0
Постављена резервна копија базе података	2	0

#### **1.2. Комуницирање са сарадницима и клијентима**

<b>ИНДИКАТОРИ</b> (максималан број бодова 8)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Прикупљени и анализирани захтеви корисника	2	0
Презентовање идејног решења (тј. плана активности)	2	0
Демонстрирање функционалности реализованог програмског решења	4	0

### **2. Израда десктоп апликација**

#### **2.1. Израђивање решења дизајна десктоп апликације**

<b>ИНДИКАТОРИ</b> (максималан број бодова 20)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Креиране форме апликације са потребним контролама	2	0
Постављени иницијални подаци у контроле на форми /формама	5	0
Подешен приказ података уз поштовање принципа прегледности и уредности форме	4	0
Извршена заштита апликације на нивоу форме	4	0

#### **2.2. Повезивање десктоп апликације са базом података**

<b>ИНДИКАТОРИ</b> (максималан број бодова 25)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Приказани подаци добијени упитом над једном или више табела у бази података	15	0
Остварена функционалност уписа, измене и/или брисања података у бази података	10	0
Извршено освежавање приказа на форми у складу са променом у бази	10	0

#### **2.3. Тестирање функционалности десктоп апликације**

<b>ИНДИКАТОРИ</b> (максималан број бодова 8)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Обезбеђена валидација унетих података	4	0
Реализовано приказивање упозорења у случају настале грешке	2	0
Извршен одабир примера за тестирање којима демонстрира заштиту апликације од грешака	2	0

### **3. Израда и управљање базом података**

#### **3.1. Модификовање структуре базе података у складу са захтевима корисника**

ИНДИКАТОРИ (максималан број бодова 10)	ПРАВИЛНО	НЕПРАВИЛНО
Анализира захтев за проширење базе и црта изменуен ЕР дијаграм базе	3	0
Креира нову табелу и повезује је са постојећим табелама у бази	2	0
Додаје нову колону постојећој табели у бази	2	0
Додаје ограничење над табелом у бази	3	0

#### **3.2. Манипулација подацима у бази података**

ИНДИКАТОРИ (максималан број бодова 15)	ПРАВИЛНО	НЕПРАВИЛНО
Креирани прости упити за претрагу података по задатим критеријумима	2	0
Креирани упити за селекцију података над више табела	3	0
Креирани командни упити за упис, измену и/или брисање података из табеле	3	0
Тестира функционалност упита	2	0

### **4. Израда техничке документације**

#### **4.1. Израђивање делова техничке документације**

ИНДИКАТОРИ (максималан број бодова 8)	ПРАВИЛНО	НЕПРАВИЛНО
Израђен технички документ (упутство) о функционалности апликације	4	0
Постављено корисничко упутство на посебну форму / веб страницу	4	0

## ОБРАЗАЦ ЗА ОЦЕЊИВАЊЕ РАДНОГ ЗАДАТКА Б

Шифра радног задатка	
Број – шифра радног налога	
Назив радног задатка	
Назив школе	
Седиште	
Образовни профил	
Име и презиме кандидата	
Име и презиме ментора	

ЗБИР БОДОВА ПО АСПЕКТИМА РАДНОГ ЗАДАТКА:							Укупно бодова
Аспекти	1.1	1.2	2.1	2.2	2.3	3.1	
Бодови							
Аспекти							
Бодови							

Члан испитне комисије:	Место и датум:
------------------------	----------------

### КОМЕНТАРИ:

## **ЗА СВАКИ ИНДИКАТОР ЗАОКРУЖИТИ ОДГОВАРАЈУЋИ БРОЈ БОДОВА**

### **1. Припрема и организација рада**

#### **1.1 Припремање и одржавање радног места**

<b>ИНДИКАТОРИ</b> (максималан број бодова 4)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Извршен избор и подешавања радног окружења	2	0
Преузети потребни ресурси	2	0

#### **1.2 Комуницирање са сарадницима и клијентима**

<b>ИНДИКАТОРИ</b> (максималан број бодова 8)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Прикупљени и анализирани захтеви корисника	2	0
Презентовање идејног решења (тј. плана активности)	2	0
Демонстрирање функционалности реализованог програмског решења	4	0

### **2. Израда веб апликација (статичких и динамичких веб страница)**

#### **2.1 Израђивање дизајнерског решења веб апликације**

<b>ИНДИКАТОРИ</b> (максималан број бодова 30)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Креиране веб странице према захтеву	8	0
Постављен одговарајући садржај на странице	5	0
Извршено стилизовање постављеног садржаја према опису у захтеву	12	0
Обезбеђено функционисање свих веза између страница	5	0

#### **2.2 Програмирање функционалности веб апликације**

<b>ИНДИКАТОРИ</b> (максималан број бодова 35)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Реализована тражена функционалност приказивања и/или измена садржаја извора података	7	0
Постављени иницијални подаци у контроле веб странице	7	0
Реализовано исправно функционисање свих контрола на веб страници према захтеву	14	0
Обезбеђено правилно функционисање свих линкова у апликацији	4	0
Веб апликација постављена на веб сервер	3	0

#### **2.3 Организовање података веб апликације**

<b>ИНДИКАТОРИ</b> (максималан број бодова 15)	<b>ПРАВИЛНО</b>	<b>НЕПРАВИЛНО</b>
Ресурси веб апликације (слике, фајлови...) су смештени у посебне фолдере	5	0
CSS код смештен у посебном фајлу	5	0
Код за функционалност и изглед организован у одговарајућим датотекама	5	0

### 3. Израда техничке документације

#### 3.1 Израђивање делова техничке документације

ИНДИКАТОРИ (максималан број бодова 8)	ПРАВИЛНО	НЕПРАВИЛНО
Израђен технички документ (упутство) о функционалности апликације	4	0
Постављен технички документ на посебну форму / веб страницу	4	0

РАЦИЈА ВЕРЗИЈА