

ДОМАЋИ ИЗ ПРЕДМЕТА КОНКУРЕНТНО И ДИСТРИБУИРАНО ПРОГРАМИРАЊЕ ЗА ШКОЛСКУ 2020/2021 У ЈУНСКОМ И ЈУЛСКОМ ИСПИТНОМ РОКУ

Пројекат из предмета 13E113КДП и 13C113КДП се у се у јунском и јулском испитном року школске 2020/2021 ради самостално.

Пројекат се ради у програмском језику Јава. Пројекат носи не више од 20 поена.

Предуслови за успешну одбрану пројекта су:

1. Уписана одговарајућа година ЕТФ одговарајућег смера.
2. Благовремена достава писаних материјала и електронске верзије решења (најмање 3 дана пре испита у испитном року или по упутству послатом на листу предмета).
3. Благовремено припремљени услови за неометану проверу рада програма у лабораторији катедре за РТИ ЕТФ-а (уколико је потребно, барем три дана пре одбране потребно је инсталирати одговарајуће програме у договору са дежурним лаборантом).
4. Успешно обављено усмено одбрана рада.

Усмена одбрана рада се састоји из следећег:

1. Кандидат који брани пројекат мора самостално да преведе и инсталира све потребне програме везане за приложено решење.
2. Кандидат мора да поседује потребан ниво знања о задатку.
3. Кандидат мора да буде свестан недостатака предложеног решења и могућности за њихово превазилажење.
4. Кандидат треба да тачно одговори на потребан број питања која се баве тематиком везаном за задатак.

Израда писаних материјала везаних за пројекат подразумева поштовање одговарајуће форме. Према тој форми сваки пројекат треба да има следеће елементе:

1. Насловну страну са јасно израженим обележјима који карактеришу овај факултет и овај предмет. Мора да садржи назив и лого факултета, назив предмета из кога се пројекат брани, назив задатка који се ради, пуно име и презиме аутора, као и број индекса, датум када је начињена прва верзија, датум када је настала текућа верзија и место где је одбрана вршена.
2. На првој страници после наслова рада и имена аутора следи садржај на српском језику писан курзивом - ***Italic*** фонтом *Times New Roman 10 pt* ћириличним писмом.
3. На наредној страни треба да се налази текст задатка који се ради. Уколико текстом задатка нешто није било довољно јасно назначено посебно уоквирити делове који су додати. Уколико предложено решење поседује изванредан број недостатака, њих назначити на посебан и лако уочљив начин, и предложити алтернативно решење које би отклонило наведене недостатке.
4. У наставку је потребно дати детаљан опис предложеног решења и свих његових карактеристика (овде није потребно стављати имплементационе детаље већ функционалне који су од суштинске важности за разумевање пројекта).
5. Након функционалне спецификације потребно је дати детаљан опис пакета, класа, интерфејса, функција и параметара, користећи **UML** спецификацију за опис интеракције (**дијаграм интеракције**).
6. Упутство за коришћење насталог програмског пакета као целине. Упутство треба да покрива два типа коришћења програмског пакета:

- а) коришћење у регуларним ситуацијама од стране особе чији је ниво рачунарског знања минималан, а која нема претходно искуство у раду са сличним пакетима
 - б) коришћење насталог програмског пакета особе чији је ниво познавања потребних вештина на задовољавајућем нивоу у циљу даљег усавршавања система.
7. Листинг програма са потребном количином коментара није потребно предавати у штампаном већ у електронском облику.
 8. Примери рада програма у регуларним и ванредним ситуацијама, са потребним објашњењима.
 9. Рад писати на српском језику уз чување оригиналних енглеских термина.

Оригинал рада треба да буде откуцан само са једне стране листова А4 формата (210 x 297 mm). Користити маргине: **2.5 cm** горња, **2 cm** доња, лева и десна. Рад треба буде писан ћириличним писмом уз коришћење фонта *Times New Roman*, величина писма: 10 типографских тачака (*10 pt*) у две колоне размакнуте **0,5 cm** уз поравнање типа *justify*. Рад куцати обичним проредом и двоструким проредом између пасуса. Почетак пасуса куцати од почетка колоне. Поднаслов у раду писати масним словима (**Bold**) великим словима величина писма: 12 типографских тачака. Сва слова у раду треба да буду црна, а позадина бела. Све табеле и слике треба да имају одговарајући наслов и да буду нумерисане. Бројеви и наслови табела налазе се увек изнад табела. Бројеви и наслови слика налазе се увек испод слика. Нумерација страна се пише у доњем десном углу. Насловна страна се не нумерише. Насловна страна треба буде писан ћириличним писмом уз коришћење фонта *Arial* у једној колони уз поравнање типа *center*. Насловна страна садржи: Назив Универзитета (величина писма: 16 типографских тачака); Назив факултета (величина писма: 16 типографских тачака); Име, средње слово, презиме и број индекса студента (величина писма: 16 типографских тачака); Наслов пројекта (величина писма: 22 типографске тачке); Предмет из кога се пројекат ради (величина писма: 16 типографских тачака); Место, година (величина писма: 14 типографских тачака).

НАПОМЕНА: Непоштовање горе наведених правила вуче умањење освојеног броја поена на усменој одбрани пројекта или у потпуности забрањује исту.

Пројекат јун 2020/2021

Берзанско пословање

Пројектовати дистрибуирани рачунарски систем који треба да омогући берзанске трансакције. Програм треба да ради у систему који се састоји од више рачунара повезаних у LAN (Local Area Network) или WAN (Wide Area Network).

У систему постоји три типа програма:

1. Централни сервер који служи за пријем, праћење цене и власништва над хартијама од вредности, као и за објављивање тренутног стања.
2. Подсервери на којима се обављају конкретне берзанске трансакције, односно куповина, продаја и израчунавање нове цене хартија од вредности.
3. Клијентски програм преко кога клијент прати тренутне цене хартија од вредности, издаје захтеве за куповином и продајом појединих.

Централни сервер својим претплаћеним корисницима нуди услуге праћења цене хартија од вредности. Ова услуга је таква да сервер периодично шаље свим претплаћеним клијентима тренутну цену свих хартија од вредности. Сервер шаље податке један по један, тако да назначавача име хартије, тренутну цену и промену у процентима у односу на претходну вредност (не шаље се податак о свакој појединачно издатој акцији/обвезници него о хартијама које се везују за неку фирму). Ово је поступак који се периодично понавља, а та периода износи x секунди. Централни сервер податке о цени добија од подсервера који воде рачуна о појединим хартијама од вредности. Централни сервер податке периодично прикупља са подсервера, при чему је тај период износи y секунди. За сваку врсту хартија од вредности постоји тачно један подсервер који води рачуна о њој. Уколико подсервер није доступан, покушава поново за $y/2$ секунди. Ако три пута за редом податак не успе да се прибави, подсервер се проглашава недоступним, треба урадити балансирање свих клијената на све доступне подсервере (*load balancing*) и све нове трансакције се пребацују на неки други подсервер коме се додељује да води евиденцију о тој хартији од вредности, док се податак о хартији задржава на последњој прочитаној вредности.

Клијент има могућност да се претплати код сервера тако што пошаље корисничко име и шифру и периодично добија податке о ценама хартија од вредности. Осим прегледа свих цена корисник може да обавља купопродајне трансакције над хартијама од вредности. Приликом купопродајних трансакција корисник шаље захтев централном серверу у ком стоји да жели да купи/прода одређену количину хартија од вредности, као и цену по којој би трансакцију да обави. Када централни сервер прими тај захтев, прослеђује га подсерверу који је задужен за те хартије. Тај подсервер од тог тренутка постаје надлежан за ту трансакцију. Уколико у тренутку приспећа захтева за куповину хартије постоји довољна количина хартија по цени која није већа од оне коју је корисник задао, обавља се трансакција. Ако не постоји онда се чека. Уколико постоји већи број које могу бити купљене, прво се гледају оне које имају најнижу цену, а ако има више оних са најнижом ценом, онда се међу њима прво узима онај који је пре приспео. Уколико у тренутку приспећа захтева за продајом хартије постоји довољна количина хартија по цени која није мања од оне коју је корисник задао, обавља се трансакција. Ако не постоји онда се чека. Уколико постоји већи број које могу бити продате прво се гледају оне које имају највишу цену, а ако има више оних са највишом ценом, онда се међу њима прво узима онај који је пре приспео.

Формирање цене поједине хартије коју је потребно проследити централном серверу када је он буде тражио се обавља на следећи начин: Ако је у интервалу између два читавања било трансакција, онда је цена специфицирана последњом обављеном трансакцијом. Ако није било трансакција него је било само понуда за продајом, узима се најнижа активна понуђена цена. Уколико је било само понуда за куповином, узима се највиша активна понуђена цена. Уколико нема активних понуда, онда се узима цена последње обављене трансакције. Уколико се корисник предомисли, може да одустане од затражене трансакције, али понуда мора да буде активна барем z секунди ($z > 2 * x$, $x > y$). Када се обави трансакција или отказивање корисник добија повратну информацију о томе.

Уколико листа подсервера која се води на централном серверу остане празна, централни сервер јавља грешку и обуставља све трансакције док се не појави барем један нови подсервер. Да би се подсервер пријавио у листу активних подсервера на централном серверу мора да пошаље централном серверу захтев за комуникацију. Том приликом централни сервер датом подсерверу додељује за које је хартије задужен. Да би балансирао оптерећење централни сервер може са једног подсервера да на други пребаци операције над неком хартијом од вредности, али тако да у било ком тренутку максимално један води рачуна о једној врсти хартија. Ни корисници ни подсервери не морају да буду у сталној вези са централним сервером. Веза може бити раскинута гашењем програма или затварањем комуникационог канала. Када се кориснички програм повеже може да тражи од сервера статусе свих трансакција. Треба обезбедити да сваки сервер може да у паралели да прима захтеве за већим бројем хартија од вредности.

Проблем решити користећи мрежну комуникацију у програмском језику Јава. За сваки од ова три типа рачунара треба да постоји одговарајући графички кориснички интерфејс (GUI треба да буде развијен користећи Java **SWING** компоненте или JavaFX). Графички кориснички интерфејс мора бити интуитиван. IP адресе и портови потребни за функционисање програма (као и остали подаци који се могу динамички мењати) морају се уносити преко графичког интерфејса (не смеју бити унапред дефинисани у изворном коду). За сваки од могућих случајева коришћења система поруке и обавештења морају бити прецизно приказани. Потребно је да се кроз графички кориснички интерфејс прикажу детаљни логови, како за централни сервер, тако и за подсервере. Сервери треба да имају могућност покретања и без корисничког интерфејса.