

Лабораторијска вежба број 2 из Објектно оријентисаног програмирања II

У сваком задатку:

- Грешке пријављивати изузецима типа класа које садрже текст поруке.
- На располагању стоји класа `Citaј` у пакету `usluge`.

1) Написати на језику *Java* следећи пакет типова:

- **Странка** банке уплаћује (>0) или подиже (<0) случајан износ новца (у опсегу од -1000 до $+1000$) који може да се дохвати.
- Кроз **активан улаз** банке, кад је отворен, странке улазе у случајним временским интервалима од 1 до 3 s. За сваку странку која уђе у банку, улазак се региструје код банке. Свака странка стане у ред код шалтера испред којег чека најмањи број странака. Улаз може да се отвори, затвори и уништи.
- Испред **активног шалтера** банке може да чека произвољан број странака који се опслужују по редоследу пристизања. Може се добити информација о броју странака у реду испред шалтера. Ако у банци нема довољно новца странка се одбија. Опслуживање странака траје случајно време од 3 до 5 s и састоји се од исписивања броја шалтера, износа новца који се уплаћује/подиже и да ли је странка опслужена или је одбијена. За сваку странку која заврши рад на шалтеру се код банке региструје излазак. Сав новац се налази на једном месту у банци. Нит шалтера се блокира ако испред нема странака.
- **Банка** има један улаз и два шалтера. Може да се региструје улазак и излазак странке, да се дохвати тренутни број странака и количина новца у банци. Приликом отварања банке задаје се и исписује почетна сума новца у банци и отвори се улаз. Приликом затварања банке затвори се улаз у банку, сачека се да све странке напусте банку и испише се преостала сума новца у банци.

Написати на језику *Java* **програм** који прочита почетну суму новца у банци и дужину радног времена, отвори банку, сачека крај радног времена, затвори банку по истеку радног времена и понавља претходне кораке док за дужину радног времена не прочита негативну вредност.

2) Написати на језику *Java* следећи пакет типова:

- **Вредносно** је нешто чему може да се одреди вредност.
- **Јединица** мере има ознаку која може да се дохвати. Дозвољене ознаке су: "ком", "l", "m" и "kg". Грешка је ако се покуша направити јединица мере с другачијом ознаком. Текстуални опис јединица мере садржи ознаку јединице.
- **Артикал** има назив и јединицу мере који могу да се дохвате. Два артикла се сматрају истим ако имају исти назив. Текстуални опис артикла садржи назив артикла.
- **Млеко** и **шећер** су артикли који се мере у литрима, односно у килограмима.
- Вредносни **запис** о артиклу има артикал, количину и јединичну цену артикла. Могу да се дохвате поља записа, да се промени количина, да се промени јединична цена и да се израчуна вредност укупне количине артикла у запису. Текстуални опис записа садржи артикал, количину, јединицу мере, јединичну цену и вредност артикла.
- **Низ** записа може да садржи задат број записа. Ствара се празан, задатог капацитета, после чега се записи могу додавати један по један (капацитет низа се по потреби аутоматски повећава за 10%, али најмање за 5 места). Може да се дохвати број елемената у низу, да се дохвати елемент са задатим редним бројем (грешка је ако је индекс изван опсега) и да се састави текстуални опис садржаја низа, по један елемент у сваком реду, при чему се текстуални описи елемената састављају **конкурентно**.
- **Складиште** има назив и адресу и садржи низ записа о артиклима. Ствара се празно, почетног капацитета од 5 записа, после чега се записи додају један по један. Грешка је ако већ постоји запис за исти артикал. Може да се дохвати запис за артикал задат по називу (грешка је ако такав запис не постоји) и да се израчуна укупна вредност артикала у складишту.

Написати на језику *Java* **програм** који направи једно складиште, стави у њега неколико артикала и испише на главном излазу укупну вредност артикала у складишту. Користити константне податке (не треба ништа учитавати).

3) Написати на језику *Java* следећи пакет типова:

- **Производ** има јединствен, аутоматски генерисан целобројан идентификатор и задату масу (g) која може да се дохвати. Може да се састави текстуални опис у облику *ид(маса)*.
- **Активан радник** има име и производи предмете задатом продуктивношћу (g/s). Може да му се зада захтев да произведе један производ задате масе који по завршеној производњи чува код себе док се не преузме од њега. Покушај задавања новог захтева док производ по претходном захтеву није завршен и преузет, или покушај преузимања производа док још није готов зауставља нит која поставља захтев док се не испуне услови за наставак рада. Може да се испита да ли постоји готов производ за преузимање и да се састави текстуални опис једног од облика: *име/сека* – ако чека на захтев за производом, *име/radi* – ако је производња у току, односно *име/производ* – ако постоји готов производ (*производ* је текстуални опис готовог производа).

Написати на језику *Java* интерактиван **програм** (с менијем) који може да извршава следеће операције (*занемарити чињеницу да код неких секвенци захтева програм може да се блокира*):

- направи радника задатог имена и продуктивности,
- захтевај од радника да направи предмет задате масе,
- испитај да ли радник има готов производ,
- преузми производ од радника и испиши производ на главном излазу,
- испиши радника на главном излазу,
- заврши програм.

4) Написати на језику *Java* следећи пакет типова:

- **Аутомобил** има јединствен, аутоматски генерисан целобројан идентификатор, задат капацитет резервоара и тренутну количину горива. Сви подаци могу да се дохвате. Почетна количина горива је случајна вредност између 10% и 30% капацитета резервоара. У аутомобил може да се сипа задата количина горива. Грешка је ако се резервоар препуни (тада се резервоар напуни и пријави грешка). Текстуални опис је облика *ид(гориво/капацитет)*.
- **Активан аутопут** има задату бензинску станицу која може да се дохвати. У случајним временским интервалима од 0,5 s до 1 s ствара по један аутомобил капацитета резервоара 50 l који додаје тој станици. Може да се прекине рад аутопута када се прекида и рад његове станице.
- **Активна пумпа** се ствара за задату бензинску станицу. Пумпа циклички дохвата по један аутомобил из реда своје станице и сипа му потребну количину горива до пуног резервоара, брзином од 1 l на сваких 100 ms. Завршетак сипања дојави бензинској станици. Текстуални опис садржи текстуални опис аутомобила којег управо опслужује.
- Бензинска **станица** има четири пумпе и ред за чекање за највише 20 аутомобила. Станица може да се отвори и затвори, може да се прекине њен рад, да јој се дода задати аутомобил на крај реда, да се извади први аутомобил из њеног реда и да јој се дојави завршетак пуњења једног аутомобила. Ако је станица затворена или је ред пун, додавање аутомобила се занемари. Ако је ред празан, при узимању се сачека да се појави неки аутомобил. Приликом затварања, прекида се чекање возила у реду и чека се да се заврши сипање горива које је у току. Приликом прекидања рада станице прекида се и рад свих њених пумпи. Текстуални опис станице садржи текстуалне описе њених пумпи и низ идентификатора аутомобила који чекају на пумпе.

Написати на језику *Java* **програм** који направи једну бензинску станицу, отвори станицу, сваке 3 s испише станицу, после 6 исписа затвори станицу и на крају је уништи. Користити константне податке (не треба ништа учитавати).

НАПОМЕНЕ:

- а) Потребно је решавати искључиво задатак чији се број добије на почетку вежбе.
- б) За израду лабораторијске вежбе, на располагању је **120** минута.
- в) Дозвољено је коришћење оригиналних књига и збирки задатака (не фотокопије).
- г) Није дозвољено коришћење унапред припремљених решења у било којем облику. Студент који користи унапред припремљена решења, биће удаљен уз анулирање поена на свим лабораторијским вежбама.
- д) У току израде лабораторијске вежбе, дежурни може студентима да постаља питања у вези њихових решења, што може утицати на број освојених поена на лабораторијској вежби.
- ђ) Студент може бити позван на накнадну одбрану рада, која може да утиче на број поена. Непојављивање студента на одбрани или показивање вишег степена неразумевања сопственог решења повлачи анулирање поена на свим лабораторијским вежбама.
- е) Резултат рада мора бити у *.java датотекама на диску L.
- ж) Оцене радова биће објављене на *Web*-у на адреси: home.etf.rs/~kraus/ (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).