ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за СИ Београд 05. и 06.05.2008.

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 `Citaj` у пакету `usluge`.

1) Написати на језику *Java* следеће типове:

- ***Мерљивим*** појмовима може да се одреди тежина.
- Мерљива ***особа*** има име и тежину. Може да се дохвати име и да се састави текстуални опис у облику *име*`(`*тежина*`)`.
- Мерљив теретни ***контејнер*** има јединствен, аутоматски генерисан регистарски број и тежину када је празан. У контејнер је могуће сместити товар задате тежине и извадити товар задате тежине. Може да се дохвати регистарски број и сопствена тежина и да се састави текстуални опис у облику *регБрој*`(`*укупнаТежина*`)`.
- Апстрактан ***авион*** има ознаку, максималну тежину, тежину када је празан и садржи низ од задатог броја мерљивих појмова. Може да се стави неки појам на задато место у низу, да се уклони појам са задатог места, да се израчуна тренутна тежина авиона и да се састави текстуални опис у облику *ознака*`(`*тренутнаТежина*`)[`*појам*`,…,`*појам*`]`. Грешка је ако се покуша претоварити авион, ставити нешто на попуњено место или уклањати нешто с празног места.
- ***Путнички*** авион може да превози само путнике, а ***теретни*** авион може да превози само контејнере. Грешка је ако се покуша додати појам неодговарајуће врсте.
- ***Аеродром*** има назив и садржи одређен број авиона у сваком тренутку. Ствара се празан, а затим авиони могу долетати и одлетати по редоследу долетања. Може да се састави текстуални опис аеродрома, тако што се у једном реду испише назив аеродрома, а затим у потребном броју редова садржани авиони. Текстуални описи садржаних авиона се састављају конкурентно.

Написати на језику *Java* програм који направи аеродром и дода на њега један путнички авион са три путника и један теретни авион са два контејнера, све са константним параметрима (не треба ништа учитавати) и после испише аеродром на главном излазу.

2) Написати на језику *Java* следеће типове:

- ***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 (с менијем) 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.

3) Написати на језику *Java* следеће типове:

- ***Производ*** има јединствен, аутоматски генерисан идентификациони број и задату масу (g) која може да се дохвати. Може да се састави текстуални опис у облику *идБрој***(***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***/ceka** – 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:

- 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* следеће типове:

- Апстрактна ***пошиљка*** има јединствен, аутоматски генерисан идентификациони број и реалну тежину који могу да се дохвате. Може да се састави текстуални опис пошиљке у облику *id***(***tez***)**.
- ***Приоритетним*** стварима може да се одреди целобројни приоритет, који може имати следеће вредности: *најнижи*, *низак*, *нормалан*, *висок* и *највиши*.
- Апстрактна ***приоритетна пошиљка*** је пошиљка с приоритетом.
- ***Писмо*** је приоритетна пошиљка најнижег приоритета. Грешка је ако тежина прелази 500g. Текстуални опис је **P***prio***[***id***(***tez***)]**.
- ***Пакет*** је приоритетна пошиљка задатог приоритета. Грешка је ако тежина прелази 50kg. Текстуални опис је **PK***prio***[***id***(***tez***)]**.
- ***Приоритетни ред*** приоритетних пошиљки се ствара празан, задатог капацитета, после чега се пошиљке додају и узимају једна по једна по нерастућем приоритету. Може да се дохвати број пошиљки у реду, да се одреди укупна тежина свих пошиљки у реду и да се састави текстуални опис реда који садржи текстуалне описе садржаних пошиљки, једна пошиљка по реду. Ако се ред препуни или ако се покуша извадити из празног реда нит извршиоца операције се привремено блокира.
- Активан ***пошиљалац*** сваке секунде генерише пошиљку случајне врсте и смешта је у задати приоритетни ред. У 20% случајева пошиљка је писмо случајне тежине од 20g до 550g. У осталим случајевима пошиљка је пакет случајне тежине од 1kg до 55kg и случајног приоритета од ниског до највишег.
- Активан ***прималац*** у случајним временским интервалима од 0,8s до 1,2s дохвата и исписује по једну пошиљку из задатог приоритетног реда.

Написати на језику *Java* програм који с главног улаза учита трајање симулације и капацитет приоритетног реда, направи један приоритетни ред, пошиљаоца и примаоца и покрене симулацију.

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 произвољних оригиналних књига и збирки задатака (не фотокопија и свезака).

г) Није дозвољено коришћење унапред припремљених решења у било којем облику. Студент који користи унапред припремљена решења, биће удаљен и **губи право на полагање колоквијума**.

д) У току израде лабораторијске вежбе, дежурни може студентима да поставља 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 и колоквијумима.

е) Текст решења распоредити у `*.java` датотеке (не стављати цело решење у једну датотеку) на диску `L`.

ж) Оцене радова биће објављене на *Web*-у на адреси: `kondor.etf.bg.ac.yu/~kraus/` (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).