

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* следећи пакет типова:

- **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 целобројан 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* следећи пакет типова:

- **Клијент** има јединствен, аутоматски генерисан целобројан идентификатор и целобројан код врсте тражене услуге у опсегу од 1 до 3. Може да се дохвати идентификатор и врста тражене услуге клијента. Текстуални опис је облика *ид (код)*.
- **Ред** клијената садржи највише 10 клијената. Клијент се додаје на крај реда и уклања се с почетка реда. Ствара се празан после чега клијенти могу да се додају и уклањају. Покушај додавања клијента у пун ред или уклањања из празног реда привремено блокира нит извршиоца. Након сваког додавања и уклањања из реда испишује свој текстуални опис на главном излазу у облику *клијент, ..., клијент*.
- **Активан службеник** има јединствен, аутоматски генерисан целобројан идентификатор и једнословну ознаку врсте. Може да обавља одређену понављајућу активност. Резултат активности је обрађивани клијент. Након сваке извршене активности испише свој текстуални опис на главном излазу у облику *врста/ид/клијент* и чека случајно време у задатом опсегу пре него што понови активност. Могуће је покренути, привремено зауставити и дефинитивно обуставити рад службеника.
- **Портир** је службеник који "ствара" клијента који тражи случајну врсту услуге с подједнаким вероватноћама и смешта га у задати улазни ред. **Разводник** је службеник који преузима клијенте из задатог улазног реда и у зависности од врсте тражене услуге ставља их у један од задатих шалтерских редова. **Шалтерски службеник** опслужује и уклања клијенте из задатог

шалтерског реда. Одговарајући редови се задају при стварању појединог службеника. Ознаке врсте службеника су, редом: **P**, **R** и **S**.

- Шалтерска **служба** садржи четири реда, један улазни и три шалтерска. Запошљава једног портира, једног разводника и три шалтерска службеника. Трајање чекања код портира је случајан временски интервал између 100 и 300 ms, код разводника између 150 и 250 ms а код шалтерског службеника између 200 и 1000 ms. Шалтерска служба може да се отвори, затвори и да се уништи. Портир пушта клијенте у службу само када је она отворена. При затварању службе, затечени клијенти у редовима се опслужују. Уништавање службе подразумева уништавање свих службеника, не чекајући да се затечени клијенти опслуже. Покретање службеника када раде и њихово заустављање када не раде нема никаквог ефекта.

Написати на језику *Java* **програм** који направи једну шалтерску службу, отвори службу, после 5 s затвори и после још 3 s је уништи. Користити константне податке (не треба ништа учитавати).

3) 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.

4) 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.

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: <http://rti.etf.bg.ac.rs/rti/ir2oo2/index.html>