ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за СИ

Београд, 13.04.2019.

Први домаћи задатак из Објектно оријентисаног програмирања 2

У сваком задатку:

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

НАПОМЕНЕ:

а) Домаћи задатак је намењен студентима за самосталну вежбу ради припреме за 2. лабораторијску вежбу.

б) Домаћи задатак се не оцењује, али један од задатака ће представљати основу за 2. лабораторијску вежбу; на вежби ће бити потребно реализовати неке класе, са или без модификације, из оригиналне поставке задатка, као и неке додатне класе везане за исти проблем.

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

- За ***израз*** може да се израчуна вредност реалног типа, да се створи израз који представља његов извод по задатој променљивој и да се састави текстуални опис израза.
- ***Константа*** је израз који има реалну вредност која не може да се промени после иницијализације. Извод константе је константа 0. Текстуални опис садржи вредност константе.
- ***Променљива*** је израз који има једнословно име и реалну вредност (подразумевано 0) која може да се промени после иницијализације. Извод променљиве по променљивој са истим именом је константа 1, а по било којој другој променљивој константа 0. Текстуални опис променљиве садржи име променљиве.
- ***Бинарни*** израз је израз који садржи два израза (a и b). При рачунању вредности израза вредности операнада се израчунавају конкурентно. Текстуални опис је облика **(***a*#*b***)** , где је # симбол реализоване операције.
- ***Збир***, ***разлика***, ***производ*** и ***количник*** су бинарни изрази чије су вредности $a+b$, $a-b$, ab и a/b, а изводи $a'+b'$, $a'-b'$, $a'b+ab'$ и $(a'b-ab')/b^2$. Покушај дељења нулом је грешка.

Написати на језику *Java* **програм** који направи објекат за израз $(x+3)/((x-2)(x+1))$, испише на главном излазу алгебарски облик израза и његовог извода и после табелира вредност израза и његовог извода на главном излазу за свако $-2 \le x \le 3$ с кораком 0,25.

2) 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***/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 (*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.

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

- ***Предмет*** има јединствен, аутоматски генерисан целобројан идентификатор и једнословну ознаку врсте предмета. Може да се састави текстуални опис појма који садржи ознаку врсте и идентификатор предмета.
- ***Мерљивим*** стварима може да се одреди реална величина.
- ***Правоугаоник*** је мерљив предмет задат дужинама ивица. Ознака врсте појма је **`P`**. Величину представља површина правоугаоника. Текстуални облик је **`P`***id*`(`*a*`,`*b*`)`.
- ***Сфера*** је мерљив предмет задат полупречником. Ознака врсте појма је **`S`**. Величину представља запремина сфере. Текстуални облик је **`S`***id*`(`*r*`)`.
- Уређен ***низ*** мерљивих ствари ствара се празан задатог капацитета после чега се ствари додају појединачно. Може да се дохвати број ствари у низу, да се извади из низа ствар задатог редног броја, да се одреди укупна величина свих ствари у низу конкурентним израчунавањем велична појединачних ствари и да се састави текстуални опис низа који садржи текстуалне описе садржаних ствари, једна ствар по реду. Грешка је ако се низ препуни или ако се покуша извадити непостојећа ствар.

Написати на језику *Java* интерактиван **програм** (с менијем) који може да извршава следеће команде: направи низ задатог капацитета, прочитај предмет и стави у низ, извади предмет из низа, испиши низ, испиши укупну величину предмета у низу и заврши с радом.

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

- Апстрактна ***пошиљка*** има јединствен, аутоматски генерисан целобројан идентификатор 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 целобројан 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 500 g. Текстуални опис је **P***prio* **[***id* **(***tez***)** **]** .
- ***Пакет*** је приоритетна пошиљка задатог приоритета. Грешка је ако тежина прелази 50 kg. 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 20 g до 550 g. У осталим случајевима пошиљка је пакет случајне тежине од 1 kg до 55 kg и случајног приоритета од ниског до највишег.
- Активан ***прималац*** у случајним временским интервалима од 0,8 s до 1,2 s 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.

5) 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 датог 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.

6) 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.

7) 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: `"kom"`, `"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).

8) 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***/***капцитет***)** .
- 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).

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

- ***Клијент*** има јединствен, аутоматски генерисан целобројан идентификатор и целобројан кôд врсте тражене услуге у опсегу од 1 до 3. Може да се дохвати идентификатор и врста тражене услуге клијента. Текстуални опис је облика *ид* **(***кôд***)**.
- ***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 исписује 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).

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

- ***Кошница*** садржи јединице меда и у њу је могуће сместити највише задат цео број јединица. Ствара се празна након чега је могуће у њу ставити наведени број јединица меда. У случају да наведени број јединица није могуће смесити, онај ко жели да стави мед чека да се ослободи довољно места. Могуће је узети сав мед из кошнице. Узимање меда је могуће само у случају када је кошница пуна. У супротном, онај ко жели да узме мед чека док се кошница не напуни медом. Могуће је дохватити тренутни број јединица меда у кошници.
- Активни ***цвет*** садржи јединице полена. Приликом стварања задаје се назив цвета и максимални број јединица полена које цвет може да произведе пре него што увене (почетни број јединица полена је 0). У случајним временским интервалима, у опсегу који се задаје приликом стварања, број јединица полена се увећава за један. Када број јединица полена у цвету достигне максимум, цвет престаје да производи полен и вене. Могуће је узети једну јединицу полена из цвета, при чему је повратна вредност индикатор успешности операције. Могуће је дохватити тренутни број јединица полена у цвету. Текстуални опис цвета је *назив* **(***полен***)** , при чему је *полен* тренутна количина полена у цвету.
- ***Ливада*** се састоји од произвољног број цветова. Ствара се празна након чега се цветови додају један по један. Могуће је дохватити случајан цвет са ливаде, при чему је грешка уколико је ливада празна. Текстуални опис ливаде је **{***цвет***,** *цвет***, . . .}**.
- Активна ***животиња*** циклички обавља одређену радњу, исписује своје име, а затим спава случајно време у опсегу задатом приликом стварања. Приликом стварања задаје се име животиње, као и број обављања задате радње пре него што животиња умре. Могуће је дохватити врсту животиње. Текстуални опис животиње је облика *име* **(***врста***)** .
- ***Пчела*** је животиња која сакупља полен и производи мед. Приликом стварања задаје се ливада на којој пчела сакупља полен, број јединица полена који је потребно да пчела сакупи да би произвела једну јединицу меда, и кошница у коју пчела смешта произведен мед. Радња пчеле се састоји од дохватања једног по једног цвета са ливаде и узимања по једне јединице полена из сваког, све док се не скупи довољна количина за производњу једне јединице меда, након чега производи мед. Након дохватања једног цвета, пчели је потребно 100ms да одлети до другог цвета са којег ће сакупљти полен.
- ***Медвед*** је животиња чија радња се састоји од узимања свог меда из задате кошнице.

Написати на језику *Java* **програм** који направи једну кошницу и једну ливаду са неколико цветова, а затим и неколико пчела и једног медведа. Покрену се животиње и цветови и програм се извршава док све животиње не угину и сви цветови не увену. Грешке пријављивати изузецима типа класа које садрже текст поруке. Користити константне податке (не треба ништа учитавати).