

## 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`.

1) 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 **Pid(a, b)**.
- **Сфера** је мерљив предмет задат полупречником. Ознака врсте појма је **S**. Величину представља запремина сфере. Текстуални облик је **Sid(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 величина 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.

2) 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).

### 3) 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 **Рprio [id (tez) ]**.
- **Пакет** је приоритетна пошиљка задатог приоритета. Грешка је ако тежина прелази 50 kg. Текстуални опис је **РКprio [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, пошиљача и примача и покрене симулацију.

### 4) 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).

#### НАПОМЕНЕ:

- а) Домаћи задатак је намењен студентима за самосталну вежбу ради припреме за 2. лабораторијску вежбу.
- б) Домаћи задатак се не оцењује.