

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* следеће типове:

- **Изразу** може да се израчуна вредност реалног типа, да се створи израз који представља његов извод по задатој променљивој и да се састави његов текстуални опис.
- **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). Текстуални опис је облика $(a\#b)$, где је $\#$ симбол реализоване операције.
- **Збир** је бинарни израз чија је вредност је $a+b$, а извод је $a'+b'$.
- **Разлика** је бинарни израз чија је вредност $a-b$, а извод је $a'-b'$.
- **Производ** је бинарни израз чија је вредност ab , а извод је $a'b+ab'$.
- **Количник** је бинарни израз чија је вредност a/b . Покушај дељења нулом је грешка. Извод је $(a'b-ab')/b^2$.

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 $-2 \leq x \leq 3$ са кораком 0,25.

2) Написати на језику *Java* следеће типове:

- **Предмет** има јединствен, аутоматски генерисан целобројан идентификатор и једнословну ознаку врсте предмета. Може да се састави текстуални опис предмета који садржи ознаку врсте и идентификатор предмета.
- **Мерљивим** стварима може да се одреди реална величина.
- **Правоугаоник** је мерљив предмет задат дужинама ивица. Ознака врсте је **P**. Величину представља површина правоугаоника. Текстуални облик је **Pid (a, b)**.
- **Сфера** је мерљив предмет задат полупречником. Ознака врсте 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. Уређивање се врши према растућој вредности величине ствари. 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 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 и испиши га, испиши низ, испиши укупну величину предмета у низу и заврши с радом.

3) Написати на језику *Java* следеће типове:

- **Пошиљка** има јединствен, аутоматски генерисан целобројан идентификатор, реалну тежину и запремину који могу да се дохвате. Може да се састави текстуални опис пошиљке у облику *id (zap , tez)* .
- **Приоритетним** стварима може да се одреди целобројни приоритет, који може имати следеће вредности: *најнижи*, *низак*, *нормалан*, *висок* и *највиши*.
- **Приоритетна пошиљка** је пошиљка с приоритетом.
- **Писмо** је приоритетна пошиљка најнижег приоритета. Грешка је ако тежина прелази 500 g и запремина 0.001 m³. Текстуални опис је **Pprio** [*id (zap , tez)*] .
- **Препоручено писмо** је писмо високог приоритета. Текстуални опис је **PPprio** [*id (zap , tez)*] .
- **Пакет** је приоритетна пошиљка задатог приоритета. Грешка је ако тежина прелази 50 kg и запремина 1 m³. Текстуални опис је **PKprio** [*id (zap , tez)*] .
- **Приоритетни ред** приоритетних пошиљки се ствара празан, задатог капацитета, после чега се пошиљке додају и узимају једна по једна. 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. Грешка је ако се ред препуни или ако се покуша извадити из празног реда.

Написати на језику *Java* интерактиван програм (с менијем) који може да извршава следеће команде: направи приоритетни ред задатог капацитета, прочитај пошиљку и стави у ред, извади пошиљку из реда, испиши ред, испиши укупну тежину пошиљки у реду и заврши с радом.

4) Написати на језику *Java* следеће типове:

- **Тачка** у равни садржи координате x и y (подразумевано $(0,0)$). Може да се дохвате координате тачке, да се израчуна растојање до задате тачке и да се састави текстуални опис у облику (x, y) .
- **Кружница** у равни садржи полупречник (подразумевано 1) и тачку која представља центар (подразумевано $(0,0)$). Могу да се дохвате центар и полупречник кружнице и да се састави текстуални опис у облику $[(x, y), r]$.
- Изломљена **линија** садржи низ тачака које чине њена темена. Ствара се празна задатог капацитета (подразумевано 5), после чега се темена додају једно по једно. Грешка је ако се низ препуни. Координате центра изломљене линије се добијају као аритметичке средње вредности x , односно y у координата темена. Може да се израчуна дужина линије и да се састави текстуални опис у облику $[t, t, \dots, t]$.
- Географски **симбол** садржи јединствен, аутоматски генерисан целобројан идентификатор. Може да се дохвати центар симбола и да се састави текстуални опис који садржи идентификатор симбола.
- **Место** је географски симбол који садржи име и кружницу. Центар симбола је центар садржане кружнице. Текстуални опис је облика **Mid: ime** $[(x, y), r]$.
- **Река** је географски симбол који садржи изломљену линију и име. Центар симбола је центар садржане линије. Текстуални опис је облика **Rid: ime** $[t, t, \dots, t]$.
- Географска **карта** може да садржи произвољан број географских симбола. Ствара се празна, после чега се географски симболи додају један по један. Може да се дохвати симбол чији је центар најближи задатој тачки и да се састави текстуални опис који се састоји од текстуалних облика садржаних симбола, један симбол по реду.

Написати на језику *Java* програм који састави пример географске карте са фиксним параметрима (није потребно читавање делова с главног улаза), испише на главном излазу састављену карту, и испише симбол у карти који је најближи фиксно задатој тачки.

5) Написати на језику *Java* следеће типове:

- **Мерљивим** појмовима може да се одреди тежина и да се дохвати назив врсте.
- Мерљива **особа** има име и тежину, који могу да се дохвате. 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)*.
- **Авион** има ознаку (низ од 5 знакова), 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 капацитет одређен максималним бројем авиона. Ствара се празан, а затим авиони могу долетати и одлетати. Покушај долетања на пун аеродром, као и одлетања са празног аеродрома, сматра се грешкама. Може да се састави текстуални опис аеродрома, тако што се у једном реду испише назив, а затим у потребном броју редова садржани авиони и то тако што се прво пише знак врсте (**P** за путнички, **T** за теретни), а затим опис авиона.

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* следеће типове:

- **Вредносној** ствари може да се одреди вредност.
- **Јединица** мере има ознаку која може да се дохвати. Дозвољене ознаке су: "cm", "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 садржај низа представи текстуално, по један елемент у сваком реду.
- **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).

7) Написати на језику *Java* следеће типове:

- Хемијски *елемент* је описан симболом елемента (до 2 знака) и редним бројем у Менделјејевом систему. Могу да се дохвате симбол и редни број елемента и може да се направи текстуални опис који садржи симбол елемента.
- *Једињење* је састављено од произвољног броја елемената. За сваки од елемената се зна број атома који улазе у састав једног молекула једињења. Ствара се празно, након чега му се додају елементи праћени податком о броју атома у молекулу. Текстуални опис једињења се састоји од низа симбола садржаних елемената праћени бројем атома за сваки елемент за који је тај број већи од 1 (на пример: NaCl , H_2SO_4).
- *Састојак* има име и садржи скуп од задатог броја једињења од којих се свако описује процентуалном количином учешћа. Ствара се празан после чега се једињења додају једно по једно (грешка је ако се препуни капацитет састојка). Састојак има количину, јединицу мере за количину, цену јединичне количине и калоријску вредност јединичне количине. Може да му се одреди укупна цена и калоријска вредност, да му се дохвати јединица мере за количину, да му се направи копија и да се направи текстуални опис који садржи име састојка, количину, јединицу мере за количину, јединичну цену и јединичну калоријску вредност.
- *Чврст састојак* је састојак чија се количина мери масом и изражава у јединицама *kg* (килограм). *Течан састојак* је састојак чија се количина мери запремином и изражава у јединицама *l* (литар).
- *Фрижидер* садржи низ састојака коначног капацитета. Ствара се празан, а састојци се додају један по један. Може да се извади састојак са задатог места из фрижидера и да се одреди укупна цена и укупна калоријска вредност састојака у фрижидеру. Покушај стављања састојка у пун фрижидер и покушај вађења састојка са непостојећег места у фрижидеру су грешке.

Написати на језику *Java* програм који направи један фрижидер, стави у њега неколико састојака и испише на главном излазу укупну вредност и укупну калоријску вредност састојака. Користити константне податке (не треба ништа учитавати).

8) Написати на језику *Java* следеће типове:

- **Предмету** може да се одреди запремина и тежина. Може да се дохвати име врсте предмета. Може да се испита сродност са другим предметом. Два предмета су сродна ако су исте врсте. Може да се састави текстуални опис који се састоји од имена класе.
- **Полупроизвод** је хомоген предмет одређене специфичне тежине веће од нуле (грешка је ако није тако). Два полупроизвода су сродна ако су исте врсте и ако су сачињени од материјала исте специфичне тежине. Текстуални опис је облика $v(\sigma)$, где су: v – име врсте и σ – специфична тежина предмета.
- **Ваљак** и **купа** су полупроизводи задатих параметара већих од нуле (грешка је ако није тако). Текстуални опис је облика $v(\sigma)(d, d, \dots, d)$, где је: d – један параметар предмета.
- **Производ** је предмет састављен од произвољног броја полупроизвода произвољне врсте. Ствара се празан после чега се саставни делови додају појединачно. Производи имају имена за идентификацију. Два производа су сродна ако имају исту врсту и садрже сродне полупроизводе по редоследу додавања. Текстуални опис је облика $v\{p, p, \dots, p\}$, где је: p – текстуални опис једног садржаног полупроизвода.
- **Складиште** може да садржи задати број (подразумевано 5) предмета. Ствара се празно. Може да се стави предмет на прво слободно место у складишту (грешка је ако нема слободног места), да се приступи предмету на месту са задим редним бројем (грешка је ако то место не постоји или ако је празно), да се дохвати капацитет складишта и да се испита да ли је неко место попуњено (грешка је ако је индекс изван дозвољеног опсега).

Написати на језику *Java* класу с функцијом која кроз дијалог прочита с главног улаза један производ и главном функцијом која направи складиште задатог капацитета, напуни складиште читајући производе с главног улаза и испише на главном излазу производ с најмањом тежином у складишту.

9) Написати на језику *Java* следеће типове:

- **Оцењивој** ствари може да се одреди целобројна оцена квалитета.
- Оцењив **испит** има задат назив предмета, оцену и датум полагања (цео број по шеми `gggmmdd`). Може да се састави текстуални опис испита у облику *назив (датум, оцена)*.
- Уређена **збирка** садржи произвољан број оцењивих ствари. Уређивање је по нерастућем редоследу оцена. Може да се дохвати број садржаних ствари, може да се уметне једна ствар, да се приступи ствари са задатим редним бројем, да се извади ствар са задатим редним бројем, да се одреди средња оцена садржаних ствари и да се састави текстуални опис збирке у облику *[ствар; ...; ствар]*.
- **Низ** је збирка која ствари складишти у облику низа. Ствара се празан, задатог почетног капацитета. Приликом уметања ствари, капацитет се по потреби повећава за 5 места. Текстуални опис на почетку садржи реч **niz**.
- **Листа** је збирка која ствари складишти у облику једноструко спрегнуте листе. Ствара се празна. Текстуални опис на почетку садржи реч **lista**.

Написати на језику *Java* програм који, читајући испите с главног улаза, направи низ испита, испише добијени низ, направи листу испита с натпросечним оценама и испише добијену листу.

10) Написати на језику *Java* следеће типове:

- **Изнајмљивој** ствари може да се одреди дневни приход (реалан број).
- Изнајмљивој **просторији** се задаје минимални и максимални број чланова који могу да је изнајме, као и основна цена изнајмљивања просторије (реалан број). Може да се изнајми на дан за задат број особа (грешка је уколико је задат број особа мањи од минималног или већи од максималног броја особа, грешка је уколико је просторија већ изнајмљена). Дневни приход просторије једнак је основној цени изнајмљивања просторије уколико је просторија изнајмљена, односно 0 уколико није изнајмљена. Може да се састави текстуални опис просторије у облику `(број_особа_за_које_је_изнајмљена, минимални_број_особа, максимални_број_особа) : приход`.
- **Соба** је просторија која има јединствен аутоматски генерисан целобројни идентификатор и реалну цену коју је неопходно платити за сваку додатну особу уколико је број особа већи од минималног броја. Цена за сваку додатну особу је јединствена за све собе и њена вредност може да се постави. Може да се израчуна дневни приход собе као сума основног прихода и износа који је неопходно платити за све додатне особе. Може да се састави текстуални опис собе у облику `Суд (број_особа_за_које_је_изнајмљена, минимални_број_особа, максимални_број_особа) : приход`.
- **Апартман** је просторија која има задати назив и задати тип опремљености. Тип опремљености може бити *Basic*, *Comfort* или *LUX*. Приход апартмана се рачуна као једном, два или три пута увећан основни приход у случају *Basic*, *Comfort* или *LUX* типа, респективно. Може да се састави текстуални опис апартмана у облику `А_назив (број_особа_за_које_је_изнајмљен, минимални_број_особа, максимални_број_особа) : приход`.
- **Низ** просторија може да садржи задат број просторија. Ствара се празан, задатог капацитета, после чега се просторије могу додавати једна по једна. Покушај додавања просторије у пун низ је без ефекта. 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 садржај низа представи текстуално тако што се у сваком реду представи једна просторија.
- **Хотел** има назив и садржи низ просторија. Ствара се са празним низом просторија, задатог капацитета, после чега му се просторије могу додавати једна по једна. Може да се одреди укупан дневни приход хотела као сума дневних прихода свих садржаних просторија. Може да се састави текстуални опис хотела у облику `Hotel_назив : приход`, након чега се у новом реду испишује низ просторија.

Написати на језику *Java* програм који, читајући податке с главног улаза, направи хотел са неколико соба и апартмана различитих типова, а затим изнајми већину соба и апартмана за по неколико особа, па испише хотел на стандардном излазу (конзоли).