

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 `Citaј` у пакету `usluge`.

1) 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 m^3 . 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 m^3 . 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.

2) 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.

3) 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.

4) 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 нарастућем 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.

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 мрежном уређају `Rad(L:)`.
- ж) Оцене радова биће објављене на *Web*-у на адреси: `home.etf.rs/~kraus/` (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).