ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за РТИ Београд 16.04.2010.

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* следеће типове:
- Хемијски ***елемент*** је описан симболом елемента (до 2 знака) и редним бројем у Мендељејевом 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`, `H2SO4`).
- ***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).
- ***Храна*** је састојак који садржи произвољан број разноврсних састојака (укључујући и друге хране). Ствара се празна, после чега се састојци додају један по један.
- ***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).

2) Написати на језику *Java* следеће типове:
- Апстрактна ***пошиљка*** има јединствен, аутоматски генерисан идентификациони број, реалну тежину и запремину који могу да се дохвате. Може да се направи копија и 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*.
- Апстрактна ***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 **P***prio*`[`*id*`(`*zap*`,`*tez*`)]`.
- ***Препоручено писмо*** је писмо високог приоритета. Текстуални опис је **PP***prio*`[`*id*`(`*zap*`,`*tez*`)]`.
- ***Пакет*** је приоритетна пошиљка задатог приоритета. Грешка је ако тежина прелази 50 kg и запремина 1 m^3. Текстуални опис је **PK***prio*`[`*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, да се направи копија реда 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.

3) 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 опси 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 направи копија, да се дохвати регистарски број и сопствена тежина и да се састави текстуални опис у облику *регБрој*`(`*укупнаТежина*`)`.
- Апстрактан ***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 садржи одређен број авиона у сваком тренутку. Ствара се празан, а затим авиони могу долетати и одлетати. 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.

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 координата темена. Може да се израчуна дужина линије и да се састави текстуални опис у облику `[`t`,`t`,…,`t`]`.
- Апстрактан географски ***симбол*** садржи јединствен, аутоматски генерисан идентификациони број. Може да се дохвати центар симбола и да се састави текстуални опис који садржи идентификациони број симбола.
- ***Место*** је географски симбол који садржи име и кружницу. Центар симбола је центар садржане кружнице. Текстуални опис је облика `M`*id*`:`*име*`[(`x`,`y`),`r`]`.
- ***Река*** је географски симбол који садржи изломљену линију и име. Центар симбола је центар садржане линије.Текстуални опис је облика `R`*id*`:`*име*`[`t`,`t`,…,`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 учитавање делова с главног улаза), испише на главном излазу састављену карту, и испише симбол у карти који је најближи фиксно задатој тачки.

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: `kondor.etf.rs/~kraus/` (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).