

## 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 `Citaaj` у пакету `usluge`.

1) 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 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 учитавање делова с главног улаза), испише на главном излазу састављену карту, и испише симбол у карти који је најближи фиксно задатој тачки.

2) Написати на језику *Java* следеће типове:

- Хемијски **елемент** је описан симболом елемента (до 2 знака) и редним бројем у Менделејејевом 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:  $\text{NaCl}$ ,  $\text{H}_2\text{SO}_4$ ).
- **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).

3) 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).

4) 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.

---

#### 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*).