

## Други домаћи задатак из Објектно оријентисаног програмирања 1

У сваком задатку, где има смисла, обезбедити конструкторе, деструктор и операторе за доделу вредности који су потребни за безбедно и ефикасно коришћење класа. Грешке пријављивати изузецима типа једноставних класа опремљених писањем поруке. Уписивање вредности објеката свих класа у излазни ток (`it<<obj`) реализовати полиморфно за полиморфне класе. Реализовати полиморфно копирање објеката полиморфних класа чији се примерци стављају у збирке. За генеричке збирке није дозвољено коришћење класа из стандардне библиотеке шаблона (*STL*).

1) Написати на језику C++ следеће класе:

- Апстрактном **предмету** се зна специфична тежина  $\sigma$  (подразумевано 1). Може да се израчуна запремина и тежина. У излазни ток се пише специфична тежина.
- **Тегла** је предмет ваљкастог облика задат полупречником основе  $r$  и висином  $h$ . У излазни ток се пише у облику  $\mathbf{T}(\sigma, r, h)$ .
- **Кутија** је предмет облика квадрата задат ивицама  $a$ ,  $b$  и  $c$ . У излазни ток се пише у облику  $\mathbf{K}(\sigma, a, b, c)$ .
- **Полица** у остави може да садржи задати број показивача на ствари неког типа. Ствара се празна са задатим бројем места (подразумевано 10) и носивости (подразумевано 100). Може да се стави ствар на прво слободно место (грешка је ако нема слободног места или ако би се полица преоптеретила), да се узме ствар са задатог места (грешка је ако место не постоји или ако је празно), да се одреди колико има слободних места на полици и да се одреди укупна тежина предмета који још могу да се додају. У излазни ток се пише у облику  $[stvar, \dots, stvar]$ .

Написати на језику C++ интерактиван програм (с менијем) за обраду једне полице за предмете са 5 места носивости 50 који може да изврши следеће команде: прочитај предмет и стави на полицу, узми предмет са прочитаног места и испиши, испиши полицу, заврши програм.

2) Написати на језику C++ следеће класе:

- Генерички **низ** може да садржи податке типа који је једини параметар шаблона. Ствара се празан задатог капацитета (подразумевано 10). Може да се дода један податак иза последњег попуњеног места, да се дохвати податак са задатим редним бројем, и да се дохвати број попуњених места. Грешка је ако се низ препуни или се покуша дохватити непостојећи податак. Низ се у излазни ток пише у облику  $\{pod, \dots, pod\}$ .
- **Мерење** температуре се задаје временом мерења и температуром, при чему је температура реалан број, а време цео број такав да се дневно време изражава у ms мерено од 00:00. Могу да се дохвате подаци о мерењу. У излазни ток се пише у облику  $(vreme, temp)$ .
- **Мерно место** садржи низ мерења обављених на једном месту. Ствара се празно задатог капацитета после чега се мерења додају једно по једно. Може да се одреди средња вредност температуре у свим садржаним мерењима. У излазни ток се пише низ мерења.
- **Мерна област** садржи низ мерних места. Ствара се празна задатог капацитета после чега се мерна места додају једно по једно. Може да се одреди мерно место с најнижом средњом температуром (грешка је ако је област празна). У излазни ток се свако садржано мерно место пише у засебном реду.

Написати на језику C++ програм који, читајући потребне податке, направи једну мерну област са 5 мерних места са по 3 мерења, испише мерну област и мерно место с најнижом средњом температуром.

3) Написати на језику C++ следеће класе:

- Апстрактна **порука** може да се упише у излазни ток. **Бројчана порука** садржи један цео број. У излазни ток се уписује садржани број. **Текстуална порука** садржи текст произвољне дужине. У излазни ток се уписује садржани текст.
- **Пошиљалац** има име које може да се дохвати. Може да се испита да ли су два пошиљаоца једнака (једнака су ако имају исто име). Пошиљалац може да се упише у излазни ток. Не сме да се прави копија пошиљаоца ни на који начин.
- **Збирка** садржи низ парова показивач на кључ неког типа и показивач на податак неког типа. Ствара се празна, задатог капацитета (подразумевано 10), после чега се парови додају појединачно (грешка је ако ако кључ већ постоји у збирци и ако се низ препуни). Може да се дохвати капацитет збирке и број попуњених места, да се испита да ли се задати кључ налази у збирци, да се дохвати показивач на податак са задатим кључем (`zbr[klj]`); грешка је ако кључ не постоји у збирци) и да се збирка упише у излазни ток, по један пар у сваком реду. Не сме да се прави копија збирке ни на који начин.
- Поштанско **сандуче** је збирка парова пошиљаоца (кључева) и порука (података).

Написати на језику C++ програм који направи једно сандуче, додаје неколико пошиљки (парова пошиљалац – порука) и дохвати и испише поруке неколико пошиљаоца. Потребне податке читати преко главног улаза.

4) Написати на језику C++ следеће класе:

- **Продавница** може да садржи задат број артикала неког типа. Ствара се празна. Може да се стави један артикал у продавницу и да се узме један артикал из продавнице по редоследу додавања. Грешка је ако се покуша стављање артикла у пуну продавницу или узимање артикла из празне продавнице. Продавница може да се упише у излазни ток у облику [*артикал, ..., артикал*].
- Апстрактан **актер** има јединствен, аутоматски генерисан целобројан идентификатор. Може да изврши неку апстрактну радњу и да се упише у излазни ток, када се пише његов идентификатор.
- **Купац** је актер чија се радња састоји од узимања из задате продавнице једног артикла неког типа уз исписивање на главном излазу свог идентификатора и иза тога узетог артикла или поруке о грешци.
- **Производ** има задат назив и цену. Може да се упише у излазни ток у облику *назив (цена)*.

Написати на језику C++ програм који направи једну продавницу и купца производа, стави неколико производа у продавницу и после изврши неколико радњи купца. Користити фиксне параметре (не треба ништа учитавати с главног улаза).

---

**НАПОМЕНЕ:**

- а) Домаћи задатак је намењен студентима за самосталну вежбу ради припреме за 3. лабораторијску вежбу.  
б) Домаћи задатак се не оцењује.