

Лабораторијска вежба број 3 из Објектно оријентисаног програмирања I

У сваком задатку где има смисла обезбедити:

- конструкторе, деструктор и оператор за доделу вредности који су потребни за безбедно коришћење класа,
- пријављивање грешака изузецима типа једноставних класа опремљених писањем поруке,
- уписивање вредности објеката свих класа у излазни ток (`it<<obj`) – полиморфно за полиморфне класе,
- 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 σ (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 полицији 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:

- Генеричка *збирка* садржи низ показивача на податке типа који је једини параметар шаблона. Ствара се празна, задатог капацитета (подразумевано 10). Може да се дода један податак иза последњег попуњеног места, да се дохвати податак са задатим редним бројем, да се дохвати број попуњених места и да се збирка упише у излазни ток. Грешка је ако се збирка препуни или се покуша дохватити непостојећи податак. Сваки податак у збирци се пише у засебном реду.
- Апстрактном *елементу* може да се одреди величина.
- *Ставка* рачуна је елемент који садржи назив артикла, јединичну цену и количину. Величину елемента представља вредност робе у ставци. У излазни ток се пишу сви садржани подаци и вредност робе.
- *Рачун* садржи име купца, датум куповине (осмоцифрен цео број по шеми `ddmmgggg`) и збирку ставки рачуна. Ствара се са празном збирком задатог капацитета после чега се ставке додају једна по једна. Може да се одреди укупна вредност купљене робе. У излазни ток се пише име купца, датум куповине, списак купљене робе и укупна вредност робе.

Написати на језику *C++* програм који читајући податке с главног улаза направи један рачун и испише га на главном излазу.

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 пише садржани број. **Текстуална порука** садржи текст произвољне дужине. У излазни ток се пише 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 по формату [*артикал*, ..., *артикал*].
- Апстрактан **актер** има јединствен, аутоматски генерисан идентификациони број. Може да изврши неку апстрактну радњу и да се упише у излазни ток, када се пише његов идентификатор.
- **Купац** је актер чија радња се 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.

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

НАПОМЕНЕ:

- Потребно је решавати искључиво задатак чији се број добије на почетку вежбе.
- За израду лабораторијске вежбе, на располагању је **120** минута.
- Дозвољено је коришћење оригиналних књига и збирки задатака (не фотокопија).
- 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.
- 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.
- 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.
- Сваку класу стављати у засебне датотеке (обавезно `.h`, по потреби и `.cpp`) и засебно програм (`.cpp`) – све на мрежном уређају `Rad(L:)`.
- Оцене радова биће објављене на *Web*-у на адреси: `home.etf.rs/~kraus/` (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).