

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:

- 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, ..., 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.

2) 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 `[artikal, ..., artikal]`.
- 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).

3) Написати на језику C++ следеће класе:

- Генеричка *листа* садржи показиваче на податке чији је тип параметар шаблона. Може да се стави податак на крај листе, и да се узме податак с почетка листе. Грешка је ако се покуша узети податак из празне листе. У излазни ток се пише у облику $\{pod, \dots, pod\}$.
- Апстрактан *производ* има јединичну цену и може да се израчуна вредност производа. У излазни ток се пише јединична цена.
- *Цигла* је производ облика квадрата задатих димензија. Вредност се рачуна као производ запремине и јединичне цене. У излазни ток се пише у облику $C(cena, a, b, c)$.
- *Складиште* садржи листу производа. Може да се стави производ у складиште, да се узме производ из складишта, да се дохвати тренутна вредност робе у складишту, да се дохвати укупна вредност робе која је прошла кроз складиште од последњег почетка посматрања промета и да се започне ново посматрање промета. У излазни ток се пише листа производа. Складиште не сме да се копира ни на који начин.

Написати на језику C++ интерактиван програм (с менијем) за обраду једног складишта који може да изврши следеће команде: прочитај производ и стави у складиште, извади из складишта и испиши производ, испиши складиште, заврши програм.

4) 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.

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 фотокопија).
- 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 нераздевања сопственог решења повлачи анулирање поена на свим лабораторијским вежбама.
- е) Сваку класу стављати у засебне датотеке (обавезно `.h`, по потреби и `.cpp`) и засебно програм (`.cpp`) – све 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*).