

Лабораторијска вежба број 2 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, као и полиморфно копирање и писање објекта полиморфних класа. Користити операторске функције где је то назначено у тексту задатка.

1) Написати на језику C++ следеће класе:

- **Квадар** се задаје реалним ивицама a , b и c (подразумевано 1, 1 и 1). Може да се израчуна запремина квадра, да се испита да ли два квадра имају једнаке ивице ($k_v1 = k_v2$) и да се квадар упише у излазни ток ($it < kv$) у облику $K(a, b, c)$.
- **Листа** квадрара може да садржи произвољан број квадрара. Ствара се празна после чега се квадрати додају један по један на крај листе ($lst += kv$). Листа може да се упише у излазни ток ($it < lst$) у облику $[k|k|...|k]$, где је k – резултат писања једног квадрара.
- **Скуп** квадрара је листа квадрара у којој су сви квадрати међусобно различити. Покушај додавања постојећег квадрара нема никаквог ефекта.

Написати на језику C++ програм који читајући податке са главног улаза направи листу и скуп квадрара, испише добијене резултате на главном излазу и понавља претходне кораке све док не прочита неки сигнал за завршетак.

2) Написати на језику C++ следеће класе:

- Апстрактно **тело** има специфичну тежину и једнословну ознаку врсте. Може да му се одреди запремина и тежина (подразумевано су једнаке нули) и да му се дохвати врста. Може да се упише у излазни ток ($it < t$).
- Дрвена **лопта** је тело задато полупречником и специфичном тежином одговарајућег дрвета. Ознака врсте је 'L'.
- Дрвени **ваљак** је тело задато полупречником основе, висином и специфичном тежином одговарајућег дрвета. Ознака врсте је 'V'.
- Дрвена **глиста** може да садржи произвољан број тела (чланака), али тако да не могу бити два чланка исте врсте један до другог. Глиста се ствара празна после чега јој се чланци додају један по један на крај ($gl += t$). Додавање погрешног чланка изазива прекид програма. Глисти може да се израчуна тежина и да се упише у излазни ток ($it < gl$) у облику $[t|t|...|t]$, где је t – резултат писања једног чланка.

Написати на језику C++ програм који читајући податке са главног улаза направи једну глисту, испише је на главном излазу, израчуна и испише њену тежину и понавља претходне кораке све док не прочита неки сигнал за завршетак.

3) Написати на језику C++ следеће класе:

- **Збирка** целих бројева може да садржи задати број елемената. Ствара се празна са задатим почетним капацитетом (подразумевано 10) и кораком повећавања капацитета (подразумевано 4). Бројеви се додају један по један на крај збирке ($z+=b$) уз повећање капацитета по потреби. Може да се дохвати број попуњених места у збирци, да се дохвати вредност задатог елемента збирке ($z[ind]$ – спречити могућност промене вредности елемента) и да се садржај збирке упише у излазни ток ($it<<z$) у облику (b, b, \dots, b) .
- **Уређена** збирка целих бројева је збирка чији су елементи уређени по неоппадајућем редоследу вредности. Нови бројеви се умећу на одговарајућа места тако да збирка остаје уређена.
- **Скуп** целих бројева је збирка чији су сви елементи међусобно различити. При додавању броја, ако исти већ постоји у скупу, не ради се ништа.

Написати на језику C++ програм који читајући податке са главног улаза направи збирку одабране врсте, испише садржај збирке на главном излазу и понавља претходне кораке све док као ознаку врсте збирке не прочита неку специјалну вредност.

4) Написати на језику C++ следеће класе:

- **Документ** садржи јединствен, аутоматски генерисан целобројан идентификатор и низ пасуса текста (један пасус је једна ниска) задатог капацитета (подразумевано 5). Ствара се празан, после чега се пасуси додају један по један ($dok+=pasus$). Може да се дохвати број пасуса, да се одреди дужина целог текста и да се документ упише у излазни ток ($it<<d$) тако што се у првом реду испише идентификатор, а затим испишу сви пасуси, са једним редом прореда између њих.
- **Ауторизован документ** је документ са потписом аутора (ниска). Потпис аутора се сматра посебним пасусом и улази у дужину текста.
- **Збирка** докумената садржи документе различитих врста. Ствара се празна, задатог капацитета, после чега се документи додају један по један ($zbr+=dok$). Може да се израчуна укупна дужина текстова свих докумената и да се упише у излазни ток ($it<<zbr$), тако што се редом испишу документи са једним редом звездица између. Збирка не може да се копира ни на који начин.

Написати на језику C++ програм који читајући податке са главног улаза направи једну збирку ауторизованих и неауторизованих докумената, затим испише на главном излазу збирну дужину текстова у збирци и понавља претходне кораке све док не прочита неки сигнал за завршетак.

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 неразумевања 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 рада биће објављене на Web-у на адреси: `home.etf.rs/~kraus/` (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).