

Лабораторијска вежба број 2 из Објектно оријентисаног програмирања I

1) Написати на језику C++ следеће класе (класе опремити оним конструкторима, деструктором и оператором за доделу вредности који су потребни за безбедно коришћење класа):

- **Вектор** у простору се задаје реалним координатама x , y и z (подразумевано $(0,0,0)$). Може да се израчуна збир v_1+v_2 и разлика v_1-v_2 вектора као и апсолутна вредност $\sim v$ вектора. Вектор може да се упише у излазни ток (`it<<v`) у облику (x,y,z) .
- **Изломљена линија** садржи низ вектора који одређују њена темена. Ствара се празна задатог максималног броја темена (подразумевано 3) после чега темена могу да се додају једно по једно (`l+=v`, ако се низ препуни, програм се прекида). Изломљена линија не сме да се копира ни на који начин. Може да јој се дохвати број темена, да јој се израчуна дужина и да се упише у излазни ток (`l.pisi(it)`) у облику $[t|t|\dots|t]$, где је t – резултат писања једног темена.
- **Многоугао** је изломљена линија код које је последње теме спојено с првим. У излазни ток се пише у облику Ml , где је l – резултат писања одговарајуће изломљене линије.

Написати на језику C++ програм који читајући податке с главног улаза направи многоугао, испише га на главном излазу, одреди његов обим и понавља претходне кораке све док за број темена не прочита недозвољену вредност.

2) Написати на језику C++ следеће класе (класе опремити оним конструкторима, деструктором и оператором за доделу вредности који су потребни за безбедно коришћење класа):

- **Тачка** у простору задаје се реалним координатама x , y и z (подразумевано $(0,0,0)$). Може да се израчуна удаљеност тачке d од координатног почетка и да се тачка упише у излазни ток (`t.pisi(it)`) у облику (x,y,z) .
- **Пондерисана тачка** у простору је тачка у простору са додатним тежинским фактором q (подразумевано 1). Удаљеност од координатног почетка се рачуна као $d \times q$, где је d – природна удаљеност тачке од координатног почетка. У излазни ток се пише у облику $(x,y,z) * q$.
- **Многоугао** садржи низ од задатог броја тачака произвољне врсте које чине темена многоугла. Ствара се празан задатог капацитета (подразумевано 3) после чега тачке могу да се додају једна по једна (`m+=t`, ако се низ препуни, програм се прекида). Многоугао не сме да се копира ни на који начин. Може да се дохвати број темена, да се пронађе теме које је најближе координатном почетку и да се многоугао упише у излазни ток (`it<<m`) у облику $[t|t|\dots|t]$, где је t – резултат писања једног темена.

Написати на језику C++ програм који читајући податке са главног улаза направи многоугао, испише га на главном излазу, пронађе и испише на главном излазу теме које је најближе координатном почетку и понавља претходне кораке све док за број темена не прочита недозвољену вредност.

3) Написати на језику C++ следеће класе (класе опремити оним конструкторима, деструктором и оператором за доделу вредности који су потребни за безбедно коришћење класа):

- **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 (класе опремити оним конструкторима, деструктором и оператором за доделу вредности који су потребни за безбедно коришћење класа):

- **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 ($t.pisi(it)$).
- 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. **Глиста не сме да се копира ни на који начин.** 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.

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) Текст решења распоредити у *.h и *.cpp датотеке (не стављати цело решење у једну датотеку).
- ж) Оцене радова биће објављене на Web-у на адреси: kondor.etf.rs/~kraus/ (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).