

Први домаћи задатак из Објектно оријентисаног програмирања 1

У сваком задатку где има смисла обезбедити конструкторе, деструктор и оператор за доделу вредности који су потребни за безбедно и ефикасно коришћење класа. Користити операторске функције где је то назначено у тексту задатка.

1) Написати на језику C++ следеће класе:

- **Квадар** се задаје ивицама a , b и c (подразумевано 1, 1, 1) које могу да се дохвате. Може да се израчуна запремина квадра, да се испита да ли два квадра имају исте ивице ($kval1==kval2$), да се испита да ли је запремина једног квадра мања од друге ($kval1<kval2$), да се квадар учита из улазног тока ($ut>>kval$) и да се квадар упише у излазни ток ($it<<kval$) у облику (a, b, c) .
- Уређени **низ** квадрара може да садржи задат број квадрара уређених по неоппадајућем редоследу њихових запремина. Ствара се празан задатог капацитета (подразумевано 5), после чега се квадрати додају један по један ($niz+=kval$; вредност функције је индикатор успеха). Низ може да се упише у излазни ток ($it<<niz$).

Написати на језику C++ програм који направи празан низ квадрара капацитета који се прочита с главног улаза, чита квадре с главног улаза и додаје низу све док додавање успева, испише на главном излазу садржај низа и понавља претходне кораке све док не прочита недозвољени капацитет низа.

2) Написати на језику C++ следеће класе:

- **Догађај** се задаје временом дешавања (час, минут и секунда, подразумевано 0,0,0; не треба проверавати исправност) и једнословном шифром догађаја (подразумевано "?"). Могу да се дохвате компоненте времена (три методе) и шифра, да се испита да ли један догађај претходи другом ($d1<d2$), да се догађај прочита из улазног тока ($ut>>d$), и да се догађај упише у излазни ток ($it<<d$) у облику **шифра [час:мин:сек]**.
- **Сценарио** представља листу догађаја уређену по неоппадајућим временима, која се ствара празна после чега се догађаји додају у листу ($scen+=t$). Може да се одреди број догађаја у сценарију и да се сценарио упише у излазни ток ($it<<scen$), један догађај по реду.

Написати на језику C++ програм који, читајући догађаје с главног улаза, направи сценарио, испише направљени сценарио на главном излазу, одреди и испише на главном излазу број догађаја у сценарију.

3) Написати на језику C++ следеће класе:

- **Мерење** температуре се задаје измереном реалном температуром (подразумевано 20°C) и реалним координатама x и y (подразумевано (0,0)) места на којем је извршено мерење. Могу да се дохвате мерно место (свака координата посебно) и температура, да се израчуна растојање између два мерна места, да се мерење прочита из улазног тока ($ut>>mer$) и да се упише у излазни ток ($it<<mer$) у облику **[температура, x , y]**.
- **Збирка** мерења садржи низ мерења. Ствара се празна задатог капацитета (подразумевано 10) после чега се мерења додају једно по једно ($zbr+=mer$; ако се низ препуни, повећава му се капацитет за 5 места). Може да се дохвати број мерења у збирци, да се приступи мерењу са задатим редним бројем ($zbr[i]$; индекс изван опсега прекида програм), да се одреди средња температура мерења која су извршена унутар круга задатог полупречника (подразумевано 1) и центра у задатој тачки (подразумевано (0,0)) и да се садржај збирке упише у излазни ток ($it<<zbr$) у облику **{ мерење, ..., мерење }**.

Написати на језику C++ програм који читајући податке с главног улаза направи збирку мерења и одреди средњу температуру унутар круга задатог полупречника око мерења задатог редног броја у збирци.

4) Написати на језику C++ следеће класе:

- **Вагон** има једнозначан аутоматски додељен целобројан идентификатор и број седишта. Вагон је већи од другог вагона (`vagon1 > vagon2`), ако је број седишта већи. Вагон може да се упише у излазни ток (`it << vagon`) у облику `ид_вагона [бр_седишта]`.
- **Композиција** вагона неограниченог капацитета се ствара празна, након чега се вагони могу додавати један по један (`kompozicija += vagon`) на крај композиције. Може да се уклони вагон с краја композиције (`kompozicija--`) или с почетка композиције (`--kompozicija`), могу да се уклоне сви вагони већи од задатог, може да се дохвати број вагона у композицији и да се композиција упише у излазни ток (`it << kompozicija`). Композиција не може да се копира ни на који начин.

Написати на језику C++ интерактиван програм (с менијем) са следећим операцијама:

- прочитај вагон с главног улаза и стави у композицију,
- избаци последњи вагон из композиције,
- избаци први вагон из композиције,
- прочитај вагон са главног улаза и избаци из композиције све вагоне веће од њега,
- испиши садржај композиције на главном излазу,
- заврши с радом.

НАПОМЕНЕ:

- а) Домаћи задатак је намењен студентима за самосталну вежбу ради припреме за 2. лабораторијску вежбу.
- б) Домаћи задатак се не оцењује.