

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. Користити операторске функције где је то назначено у тексту задатка.

1) 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.

2) Написати на језику C++ следеће класе:

- **Вектор** у простору се задаје компонентама у правцу x , y и z оса (подразумевано (0,0,0)). Могу да се дохвате компоненте вектора (три методе), да се налази вектор супротне оријентације ($-v$), да се израчуна збир два вектора ($v1+v2$, $v1+=v2$), да се израчуна производ вектора и реалног броја ($v*r$, $r*v$, $v*=r$), да се вектор учита из улазног тока ($ut >> v$) и да се упише у излазни ток ($it << v$) у облику (v_x, v_y, v_z) .
- **Многоугао** у простору садржи низ од задатог броја (подразумевано 5) вектора који представљају положаје темена. Ствара се са теменима равномерно распоређеним по јединичном кругу у равни x - y (координате темена: $(\cos i\phi, \sin i\phi, 0)$; $i=0, \dots, n-1$; $\phi=2\pi/n$) која касније могу да се промене. Може да се дохвати број темена многоугла, да се дохвати задато теме ($m[i]$ – индекс изван опсега прекида програм), да се одреди вектор положаја тежишта многоугла (аритметичка средња вредност вектора положаја темена), да се многоугао помери за задати вектор помака и да се многоугао упише у излазни ток ($it << m$) у облику $\mathbf{M}[v_0, \dots, v_{n-1}]$, где је v_i резултат исписивања вектора положаја једног темена.

Написати на језику C++ програм који прочита један многоугао с главног улаза, помери га тако да тежиште дође у координатни почетак и испише добијени многоугао на главном излазу.

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 ($it > mer$) 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:

- **Порука** се задаје текстом поруке (ниска знакова) и целобројним приоритетом (мањи број означава виши приоритет). Може да се испита да ли једна порука има виши приоритет од друге ($por1 > por2$) и да се порука упише у излазни ток ($it < por$) у облику [*текст*] :*прио* .
- **Преносна линија** може да садржи произвољан број порука. Ствара се празна после чега поруке се додају једна по једна ($lin += por$). Може да се дохвати број порука у линији, да се порука највишег приоритета извади из линије и да се линија упише у излазни ток ($it < lin$). Од две поруке једнаког приоритета прво се узима она која је пре стављена у линију. Покушај узимања поруке из празне линије прекида програм.

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

- прочитај поруку с главног улаза и стави у линију,
- узми поруку из линије и испиши на главном излазу,
- испиши садржај линије на главном излазу,
- заврши с радом.

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) Резултат рада мора бити у *.h и *.cpp датотекама у фасцикли L:\.
- ж) Оцене радова биће објављене на *Web*-у на адреси: kondor.etf.rs/~kraus/ (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).