

## 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:

- **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 ( $ut \gg mer$ ) и да се упише у излазни ток ( $it \ll mer$ ) 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 ( $it \ll zbr$ ) 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.

2) Написати на језику C++ следеће класе:

- **Студент** се задаје именом и презименом (једна ниска знакова), бројем индекса и годином уписа. Студент је иза другог студента ( $stud1 > stud2$ ), ако је година уписа већа, а за једнаку годину уписа ако је број индекса већи. Студент може да се упише у излазни ток ( $it \ll stud$ ) у облику *име\_и\_презиме*[*бр\_индекса/год\_уписа*].
- **Клуна** ограниченог капацитета је отворена само са једне стране, одакле студенти могу да улазе и седају, а са друге стране је затворена зидом, тако да први студент који је сео мора последњи да је напусти. Не може се улазити и излазити са произвољног места клупе. Клуна се ствара празна, (са празним низом показивача на студенте), после чега студенти могу да улазе и седају на њу ( $klupa \ll stud\_pokaz$ ), као и да устају и излазе из ње ( $klupa \gg stud\_pokaz$ ). Покушај уласка студента у препуну клупу, као и покушај изласка из празне клупе прекида програм. Може да се дохвати број студената у клупи, да се одреди последње уписани од студената у клупи и да се клупа упише у излазни ток ( $it \ll klupa$ ). Клуна не може да се копира ни на који начин.

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

- прочитај студента с главног улаза и стави у клупу,
- избаци првог на реду студента из клупе и испиши га на главном излазу,
- испиши последње уписаног студента из клупе,
- испиши садржај клупе на главном излазу,
- заврши с радом.

3) Написати на језику C++ следеће класе:

- **Вагон** има једнозначан аутоматски додељен целобројан идентификатор и број седишта. Вагон је већи од другог вагона (`wagon1 > wagon2`), ако је број седишта већи. Вагон може да се упише у излазни ток (`it << wagon`) у облику `ид_вагона[бр_седишта]`.
- **Композиција** вагона неограниченог капацитета се ствара празна, након чега се вагони могу додавати један по један (`kompozicija += wagon`) 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.

4) 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 квадрa, да се испита да ли два квадрa имају исте ивице (`kvad1 == kvad2`), да се испита да ли је запремина једног квадрa мања од друге (`kvad1 < kvad2`), да се квадар учита из улазног тока (`ut >> kvad`) и да се квадар упише у излазни ток (`it << kvad`) у облику  $(a, b, c)$ .
- Уређени **низ** квадрa може да садржи задат број квадрa уређених по неоппадајућем редоследу њихових запремина. Ствара се празан задатог капацитета (подразумевано 5), после чега се квадрa додају један по један (`niz += kvad`; вредност функције је индикатор успеха). Низ може да се упише у излазни ток (`it << niz`).

Написати на језику C++ програм који направи празан низ квадрa капацитета који се прочита с главног улаза, чита квадрe 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.

---

**НАПОМЕНЕ:**

- Потребно је решавати искључиво задатак чији се број добије на почетку вежбе.
- За израду лабораторијске вежбе, на располагању је **120** минута.
- Дозвољено је коришћење оригиналних књига и збирки задатака (не фотокопија).
- 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.
- 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.
- 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.
- Сваку класу стављати у засебне датотеке (обавезно `.h`, по потреби и `.cpp`) и засебно програм (`.cpp`) – све на мрежном уређају `Rad(L:)`.
- Оцене радова биће објављене на *Web*-у на адреси: `home.etf.rs/~kraus/` (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).