

## 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:

- **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.

2) 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 квадрата, да се испита да ли два квадрата имају исте ивице ( $kvad1 == kvad2$ ), да се испита да ли је запремина једног квадрата 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** квадрата може да садржи задат број квадрата уређених по неоппадајућем редоследу њихових запремина. Ствара се празан задатог капацитета (подразумевано 5), после чега се квадрати 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 квадрата капацитета који се прочита с главног улаза, чита квадрате 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.

3) Написати на језику C++ следеће класе:

- **Елемент** слике (*pixel*) се задаје помоћу целобројних интензитета црвене, зелене и плаве боје у опсегу од 0 до 255 (подразумевано (255,255,255), тј. бела боја). Могу да се дохвате компоненте боје, да се боја прочита из улазног тока (`ut>>elem`) и да се упише у излазни ток (`it<<elem`) у облику (*црвена, зелена, плава*).
- **Слика** правоугаоног облика садржи матрицу задатог броја врста и колона (подразумевано 3×4) елемената слике задате почетне боје (подразумевано беле). Могу да се дохвате димензије слике, да се приступи елементу са задатим индексима (`sli(i, j)`; индекс изван опсега прекида програм) и да се слика упише у излазни ток (`it<<sli`), свака врста у засебном реду.

Написати на језику C++ програм који читајући податке с главног улаза направи слику задатих димензија са свим белим елементима, промени боју неких елемената и испише слику на главном излазу.

4) 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<<stud`) у облику *име\_и\_презиме[бр\_индекса/год\_уписа]*.
- **Клупа** 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. Клупа се ствара празна, (са празним низом показивача на студенте), после чега студенти могу да улазе и седају на њу (`klupa<<stud_pokaz`), као и да устају и излазе из ње (`klupa>>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<<klupa`). Клупа 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.

---

#### 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) Класе стављати у засебне датотеке (обавезно `.h`, по потреби и `.cpp`) и засебно програм (`.cpp`) – све на мрежном уређају `Rad(L:)`.
- ж) Оцене радова биће објављене на *Web*-у на адреси: `kondor.etf.rs/~kraus/` (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).