

Лабораторијска вежба број 3 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,
- пријављивање грешака изузецима типа једноставних класа опремљених писањем поруке,
- уписивање вредности објеката свих класа у излазни ток (`it<<obj`) – полиморфно за полиморфне класе,
- полиморфно копирање објеката полиморфних класа чији се примерци стављају у збирке.

За генеричке збирке није дозвољено коришћење класа из стандардне библиотеке шаблона (*STL*).

1) Написати на језику *C++* следеће класе:

- Генерички **низ** може да садржи податке типа који је једини параметар шаблона. Ствара се празан капацитета 5 после чега се подаци додају један по један на крај, уз повећавање капацитета за по 5 места кад год се низ препуни. Може да се дохвати број података у низу и да се дохвати податак са задатим редним бројем. Грешка је ако се покуша дохватити непостојећи податак.
- **Датотека** има име, величину и показивач на именик (родитељску датотеку, *Folder*) који је садржи. При стварању датотеке родитељ остаје недодељен. Може да се дохвати величина, да се постави родитељ, да се ствара динамичка копија и да се упише у излазни ток у облику *назив_датотеке(величина)*, где је назив датотеке облика *назив_родитеља/име_датотеке*.
- **Именик** је датотека која садржи произвољан број датотека. Величина именика је $100+10n$, где је n број датотека у именику. Ствара се празан после чега се датотеке додају једна по једна уз постављање именика за родитеља датотеке. Грешка је ако у именику већ постоји датотека с истим именом. Може да се дохвати број датотека у именику, да се дохвати датотека са задим редним бројем и да се одреди укупна величина свих датотека у именику.
- **Диск** садржи један именик који представља корен стабла датотека. Име тог именика је празно (" "). Може да се дохвати корени именик.

Написати на језику *C++* програм који направи диск са два именика са по две датотеке и испише на главном излазу све датотеке на диску. Користити константне параметре (не треба ништа читати с главног улаза).

2) Написати на језику *C++* следеће класе:

- Апстрактном **предмету** се зна специфична тежина σ (подразумевано 1). Може да се израчуна запремина и тежина. У излазни ток се пише специфична тежина.
- **Тегла** је предмет ваљкастог облика задат полупречником основе r и висином h . У излазни ток се пише у облику "**T**(σ, r, h)".
- **Кутија** је предмет облика квадра задат ивицама a, b и c . У излазни ток се пише у облику "**K**(σ, a, b, c)".
- **Полица** у остави може да садржи задати број предмета. Ствара се празна са задатим бројем места (подразумевано 10) и носивости (подразумевано 100). Може да се стави предмет на прво слободно место (грешка је ако нема слободног места или ако би се полица преоптеретила), да се узме предмет са задатог места (грешка је ако место не постоји или ако је празно), да се одреди колико има слободних места на полици и да се одреди укупна тежина предмета који још могу да се додају. У излазни ток се пише у облику "**[predm, ..., predm]**".

Написати на језику *C++* интерактиван програм (с менијем) за обраду једне полице са 5 места носивости 50 који може да изврши следеће команде: прочитај предмет и стави на полицу, узми предмет са прочитаног места и испиши, испиши полицу, заврши програм.

3) Написати на језику C++ следеће класе:

- Генеричка **збирка** може да садржи податке типа који је једини параметар шаблона. Ствара се празна задатог капацитета (подразумевано 10). Може да се дода један податак иза последњег попуњеног места, да се дохвати податак са задатим редним бројем, да се дохвати број попуњених места и да се упише у излазни ток. Грешка је ако се збирка препуни или се покуша дохватити непостојећи податак. Сваки податак у збирци се пише у излазни ток у засебном реду.
- Апстрактном **елементу** може да се одреди величина, да се направи динамичка копија и да се упише у излазни ток.
- **Ставка** рачуна је елемент који садржи назив артикла, једничну цену и количину. Величину елемента представља вредност робе у ставци. У излазни ток се пишу сви садржани подаци и вредност робе.
- **Рачун** садржи име купца, датум куповине (осмоцифрен цео број по шеми ddmmsggg) и збирку ставки рачуна. Ствара се са празном збирком задатог капацитета после чега се ставке додају једна по једна. Може да се одреди укупна вредност купљене робе и да се рачун упише у излазни ток. У излазни ток се пишу име купца, датум куповине, списак купљене робе и укупна вредност робе.

Написати на језику C++ програм који читајући податке с главног улаза направи један рачун и испише га на главном излазу.

4) Написати на језику C++ следеће класе:

- Генерички **низ** може да садржи податке типа који је једини параметар шаблона. Ствара се празан задатог капацитета (подразумевано 10). Може да се дода један податак иза последњег попуњеног места, да се дохвати податак са задатим редним бројем, и да се дохвати број попуњених места. Грешка је ако се низ препуни или се покуша дохватити непостојећи податак. Низ се у излазни ток пише у облику "{*pod*, ..., *pod*}".
- **Мерење** температуре се задаје временом мерења и температуром, при чему је температура реалан број, а време цео број такав да се дневно време изражава у ms мерено од 00:00. Могу да се дохвате подаци о мерењу. У излазни ток се пише у облику "(*vreme*, *temp*)".
- **Мерно место** садржи низ мерења обављених на једном месту. Ствара се празно задатог капацитета после чега се мерења додају једно по једно. Може да се одреди средња вредност температуре у свим садржаним мерењима. У излазни ток се пише низ мерења.
- **Мерна област** садржи низ мерних места. Ствара се празна задатог капацитета после чега се мерена места додају једно по једно. Може да се одреди мерно место с најнижом средњом температуром. Грешка је ако је област празна. У излазни ток се свако садржано мерно место пише у засебном реду.

Написати на језику C++ програм који, читајући потребне податке, направи једну мерну област са 5 мерних места са по 3 мерења, испише мерну област и мерно место с најнижом средњом температуром.

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 и колоквијумима.
- 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) на диску L.
- ж) Оцене радова биће објављене на Web-у на адреси: kondor.etf.rs/~kraus/ (одреднице: *настава* | <име предмета> | *оцене* | *колоквијуми*).