ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
Одсек за РТИ и *Одсек за СИ* Београд 27.12.2014.

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:

- Генеричка ***тачка*** у равни садржи координате *x* и *y* неког нумеричког типа (подразумевано (0,0)). Могуће је дохватити координате тачке и израчунати растојање до задате тачке. У излазни ток се пише у облику **(***x***,***y***)**.
- Апстрактан географски ***симбол*** садржи јединствен, аутоматски генерисан идентификациони број. Могуће је дохватити центар симбола који је тачка с реалним координатама. У излазни ток се пише идентификациони број симбола.
- ***Место*** је географски симбол који садржи име и тачку с реалним координатама која представља центар места. У излазни ток се пише у облику **M***id***:***ime***[***t***]**, где је *t* – резултат писања тачке.
- Географска ***карта*** садржи произвољан број географских симбола. Ствара се празна, после чега се географски симболи могу додавати један по један. Могуће је дохватити симбол чији је центар најближи задатој тачки. Грешка је ако карта не садржи ниједан симбол. У излазни ток се сваки садржани симбол пише у засебном реду.

Написати на језику *C++* програм који читајући податке с главног улаза направи једну географску карту, испише је на главном излазу и испише симбол који је најближи тачки коју прочита с главног улаза.

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

- Генеричка ***збирка*** садржи низ показивача на 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 збирка упише у излазни ток. Грешка је ако се збирка препуни или се покуша дохватити непостојећи податак. Сваки податак у збирци се пише у засебном реду.
- Апстрактном ***елементу*** може да се одреди величина.
- ***Ставка*** рачуна је елемент који садржи назив артикла, јединичну 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 `ddmmgggg`) 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. У излазни ток се пише 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.

3) 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) која је садржи. При стварању датотеке родитељ остаје "празан". Може да се дохвати величина, да се постави родитељ, да се ствара полиморфна 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+10*n*, 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 задатим 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 на диску.

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).

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

- ***Збирка*** садржи низ елемената неког типа. Ствара се празна задатог почетног капацитета (подразумевано 10) и корака повећања капацитета (подразумевано 3), после чега се елементи додају један по један на крај низа (`niz+=pod`). У случају препуњења низа, његов капацитет се повећава за задати корак. Може да се дохвати број елемената низа, да се дохвати податак са задатим редним бројем (`niz[ind]`), да се извади из низа податак са задатим редним бројем (остали елементи попуњавају упражњено место) и да се низ испразни. Недозвољен редни број при дохватању и вађењу елемента је грешка.
- Апстрактној ***роби*** може да се одреди вредност реалног типа, да се направи њена полиморфна копија и да се упише у излазни ток (`it<<roba`).
- ***Артикал*** је роба која има ознаку од једног знака (подразумевано `'?'`) и задату вредност (подразумевано 0). Може да се дохвати ознака артикла. У излазни ток се пише ознака артикла.
- ***Пакет*** је роба која садржи збирку робе. Ствара се празан после чега се робе додају појединачно (`paket+=roba`). У излазни ток се пишу садржане робе унутар пара угластих заграда (`[]`).

Написати на језику *C++* функцију која кроз дијалог прочита једну робу и главну функцију која чита с главног улаза по једну робу и испише је на главном излазу све док не прочита "празну" робу.

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 поставља 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) Сваку класу 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: http://rti.etf.bg.ac.rs/rti/ir2oo1/index.html.