ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMSKI JEZICI I - 2. kolokvijum (10.12.2009) – RI - A

1. Definisati klasu **Element** koja predstavlja apstrakciju hemijskih elemenata. Svaki hemijski element opisan je **simbolom** (do 2 znaka), **rednim brojem** u periodnom sistemu elemenata i **masenim brojem** (približno jednako ukupnom broju protona i neutrona u jezgru atoma).

   Klasa Element treba da ima:

   a) **konstruktor** – podrazumijevano kreira element čiji je naziv "*", a redni i maseni broj nula. Inače, element može da se inicijalizuje na osnovu podataka koje konstruktor može da primi kao argumente.
   b) funkciju članicu **getSimbol** – vraća simbol elementa.

   Za klasu Element treba preklopiti sljedeće **operatore**:

   a) **operator <<** - na zadatom izlaznom medijumu ispisuje simbol elementa. Omogućiti kaskadno pozivanje.
   b) **operator ()** - vraća maseni broj elementa.

   Razdvojiti interfejs od implementacije, tj. funkcije članice definisati izvan definicije klase.

2. Definisati klasu **Molekul** koja predstavlja apstrakciju molekula neke materije. Svaka materija sačinjena je od jednog ili više različitih elemenata, pri čemu svaki molekul ima tačno određen broj atoma konkretnog elementa (npr. molekul vode (H_2O) čine dva atoma vodonika (H) i jedan atom kiseonika (O), molekul sulfatne kiseline (H_2SO_4) čine dva atoma vodonika, jedan atom sumpora i četiri atoma kiseonika, itd.).

   Klasa Molekul treba da ima:

   a) **konstruktor** – kreira prazan molekul.
   b) **konstruktor kopije**.

   Za klasu Molekul treba preklopiti sljedeće operatore:

   a) **operator <<** - na zadatom izlaznom medijumu ispisuje molekul, pri čemu se ispisuju redom simboli elemenata i broj atoma svakog elementa ako je taj broj veći od 1 (npr. H2O). Omogućiti kaskadno pozivanje.
   b) **operator +=** - omogućava da se molekulu dodaju elementi atom po atom (molekul += element).
   c) **operator ()** - vraća maseni broj molekula (zbir masenih brojeva svih atoma u molekulu).

   Razdvojiti interfejs od implementacije, tj. funkcije članice definisati izvan definicije klase.

3. Napisati program u kojem treba:
   a) kreirati konstantne elemente neophodne za formiranje sulfatne kiseline (H_2SO_4).
   b) kreirati niz od tri prazna molekula, a zatim korišćenjem operatora += formirati: H_2O, SO_2, H_2SO_4.
   c) na standardnom izlazu ispisati formirane molekule u svakom redu po jedan molekul.

4. Odrediti izlaz iz sljedećeg programa:

```
#include <iostream.h>
class X
{
  public: X(int X) { cout << X; }
          ~X() { cout << "X"; }
};
class A
{
  public: A() : x(bk) { cout << ++bk << endl; }
          ~A() { cout << ++bd << endl; }
  private: static int bk, bd;
           X x;
};
int A::bk;  int A::bd;
void f(A x) { A a=x; }
main() { A a, *pa, &rb=a; f(a); }
```

0	1						
1							
X	2						
X	3						
X							

Raspodjela bodova po zadacima

Zadatak	1	2	3	4	Σ
Bodovi	30%	45%	10%	15%	100%

Potrebni podaci iz Periodnog sistema elemenata

1	...	8	...	16
H		O		S
1,01		16,00		32,07

ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMSKI JEZICI I - 2. kolokvijum (10.12.2009) – RI - B

1. Definisati klasu **Artikal** koja predstavlja apstrakciju artikala koji se prodaju u nekoj prodavnici. Svaki artikal opisan je **nazivom** (do 20 znakova), **jediničnom cijenom** (cijena po jedinici mjere) i **količinom**.

   Klasa Artikal treba da ima:
   a) **konstruktor** – podrazumijevano kreira artikal čiji je naziv "?", a cijena i količina nula. Inače, artikal može da se inicijalizuje na osnovu podataka koje konstruktor može da primi kao argumente.
   b) funkciju članicu **readArtikal** – sa standardnog ulaza učitava podatke o artiklu.

   Za klasu Artikal treba preklopiti sljedeće **operatore**:
   a) **operator <<** - na zadatom izlaznom medijumu ispisuje podatke o artiklu u obliku: **`Naziv Kolicina Cijena Vrijednost`**, gdje vrijednost predstavlja ukupnu vrijednost artikla (količina*cijena). Omogućiti kaskadno pozivanje.
   b) **operator ()** - vraća cijenu artikla.

   Razdvojiti interfejs od implementacije, tj. funkcije članice definisati izvan definicije klase.

2. Definisati klasu **Racun** koja predstavlja apstrakciju računa koji se izdaju kupcima za kupljenu robu. Svaki izdati račun ima jedinstveni broj (prvi račun ima broj 1, drugi račun broj 2, itd). Svaki račun čini jedna ili više stavki, pri čemu svaka stavka evidentira prodaju jednog artikla.

   Jedinstvenu numeraciju računa treba realizovati korišćenjem zajedničkih članova u klasi.

   Klasa Racun treba da ima **konstruktor** (kreira prazan račun sa odgovarajućim jedinstvenim brojem) i **destruktor**.

   Za klasu Racun treba preklopiti sljedeće operatore:
   a) **operator <<** - na zadatom izlaznom medijumu ispisuje račun, pri čemu se redom ispisuju evidentirane stavke, svaka stavka u novom redu, u obliku: **`Rb. Naziv artikla Kolicina Cijena Iznos`**
      Omogućiti kaskadno pozivanje.
   b) **operator +=** - omogućava da se računu dodaju nove stavke, artikal po artikal (racun += artikal).
   c) **operator ()** - vraća ukupnu vrijednost računa (zbir vrijednosti svih stavki na računu).

   Razdvojiti interfejs od implementacije, tj. funkcije članice definisati izvan definicije klase.

3. Napisati program u kojem treba:
   a) učitati *n* (*n*>0) artikala sa standardnog ulaza.
   b) kreirati prazan račun, a zatim sve učitane artikle prodati (na račun dodati odgovarajuće stavke korišćenjem operatora +=).
   c) na standardnom izlazu ispisati formirani račun (kao u primjeru).

4. Odrediti izlaz iz sljedećeg programa:

```
#include <iostream.h>
class B
{
  public: B(int B) { cout << B; }
         ~B() { cout << "A"; }
};
class A
{
  public: A() : b(bk) { cout << bk++ << endl; }
         ~A() { cout << bd++ << endl; }
  private: static int bk, bd;
           B b;
};
int A::bk;  int A::bd;
A f(A x) { return x; }
main() { A a, *pa, &rb=a; f(a); }
```

0	**0**						
0							
A	**1**						
A	**2**						
A							

Raspodjela bodova po zadacima

Zadatak	1	2	3	4	Σ
Bodovi	30%	45%	10%	15%	100%

Primjer izvršavanja programa

```
n=2
1. artikal
     naziv: Sladoled
     cijena: 1.50
     kolicina: 1
2. artikal
     naziv: Mandarine
     cijena: 2.50
     kolicina: 2
====================================

              Racun br. 1
------------------------------------
Rb. Naziv     Kolicina Cijena   Iznos
--- --------- -------- ------ -------
 1. Sladoled      1.00   1.50    1.50
 2. Mandarine     2.00   2.50    5.00
------------------------------------
                        UKUPNO:    6.50
```

ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMSKI JEZICI I - 2. kolokvijum (09.12.2009) – ET - A

1. Definisati klasu **Tacka** koja predstavlja apstrakciju tačke u koordinatnom sistemu x0y, u kojem je svaka tačka opisana koordinatama **x** i **y**. Svaka tačka ima jedinstvenu oznaku (prva tačka ima oznaku A, druga oznaku B, itd).

   Jedinstveno označavanje tačaka treba realizovati korišćenjem zajedničkih članova u klasi.

   Klasa Tacka treba da ima:
   a) **konstruktor** – podrazumijevano postavlja tačku u centar koordinatnog sistema. Inače, tačka može da se inicijalizuje na osnovu podataka koje konstruktor može da primi kao argumente.
   b) funkciju članicu **readTacka** – sa standardnog ulaza učitava i postavlja koordinate tačke.

   Za klasu Tacka treba preklopiti sljedeće **operatore**:
   a) **operator <<** - na zadatom izlaznom medijumu ispisuje tačku u obliku: oznaka(x,y), npr. A(1.0,2.0). Omogućiti kaskadno pozivanje.
   b) **operator ()** - vraća udaljenost do neke druge tačke (podrazumijevano do centra koordinatnog sistema).

   Razdvojiti interfejs od implementacije, tj. funkcije članice definisati izvan definicije klase.

2. Definisati klasu **Poligon** koja predstavlja apstrakciju poligona u ravni x0y. Svaki poligon određen je sa tri ili više vrhova, koji predstavljaju tačke u datoj ravni.

   Klasa Poligon treba da ima **konstruktor** – može da primi koordinate svih vrhova i kreira odgovarajući poligon, i **destruktor**.

   Za klasu Poligon treba preklopiti sljedeće operatore:
   a) **operator <<** - na zadatom izlaznom medijumu ispisuje poligon u obliku: `tacka – tacka - ... - tacka` (npr. A(1.0,2.0) – B(3.0, 2.1) – C(1.0, 2.1)). Omogućiti kaskadno pozivanje.
   b) **operator +=** - omogućava da se poligonu doda tačka (poligon += tacka).
   c) **operator ()** - vraća obim poligona.

   Razdvojiti interfejs od implementacije, tj. funkcije članice definisati izvan definicije klase.

3. Napisati program u kojem treba:
   a) kreirati tri konstantne tačke A(1.0, 1.0), B(3.0, 1.0) i C(3.0, 3.0).
   b) kreirati trougao sa vrhovima A, B i C, pa ispisati kreirani poligon i njegov obim.
   c) kreiranom trouglu dodati četvrti vrh, tako da poligon predstavlja kvadrat, pa ispisati modifikovani poligon i njegov obim.

4. Odrediti izlaz iz sljedećeg programa:

```
#include <iostream.h>
class A
{
  public:
     A() { ++count; }
     ~A() { cout << --count << endl; }
     static void report() { cout << count; }
  private: static int count;
};
int A::count=5;
void f(A x) { A a=x; }
main()
{
   A a, *pa, &rb=a;
   A::report(); f(a); A::report();
}
```

6	**5**						
4							
4	**3**						

Raspodjela bodova po zadacima

Zadatak	1	2	3	4	Σ
Bodovi	30%	45%	10%	15%	100%

Primjer izvršavanja programa

```
3-ugao: A(1,1)-B(3,1)-C(3,3)
        Obim: 6.82
4-ugao: A(1,1)-B(3,1)-C(3,3)-D(1,3)
        Obim: 8
```

ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMSKI JEZICI I - 2. kolokvijum (09.12.2009) – ET - B

1. Definisati klasu **Vrijeme** koja predstavlja apstrakciju vremenskog trenutka, koji se opisuje u obliku hh:mm:ss, gdje je hh - broj sati u intervalu 0-23, mm – broj minuta u intervalu 0-59, ss – broj sekundi u intervalu 0-59.

   Klasa Vrijeme treba da ima **konstruktor** – podrazumijevano postavlja vrijeme na 00:00:00. Inače, vrijeme može da se inicijalizuje na osnovu podataka koje konstruktor može da primi kao argumente (Obratiti pažnju na korektnost podataka, npr. ako se konstruktoru pošalje 70 minuta, to znači 1 sat i 10 minuta). U slučaju da se pokuša postaviti vrijeme veće od 23:59:59, vrijeme treba postaviti na podrazumijevano 00:00:00.

   Za klasu Vrijeme treba preklopiti sljedeće **operatore**:
   a) **operator <<** - na zadatom izlaznom medijumu ispisuje vrijeme u obliku: hh:mm:ss. Omogućiti kaskadno pozivanje.
   b) **operator konverzije u tip long** – vraća cjelobrojni podatak koji odgovara ukupnom broju sekundi (npr. 1:10:20 je 1*3600+10*60+20 sekundi).
   c) **operator <** - poredi dva vremenska trenutka i vraća rezultat poređenja (tj. da li je prvi manji od drugog).
   d) **operator ==** - vraća rezultat poređenja dva vremenska trenutka (tj. da li su jednaki).

   Razdvojiti interfejs od implementacije, tj. funkcije članice definisati izvan definicije klase.

2. Definisati klasu **Raspored** koja omogućava vođenje rasporeda nastave u toku jednog dana. Svako predavanje karakterišu vrijeme početka i naziv (niz od maksimalno 20 znakova).

   Klasa Raspored treba da ima:
   a) **konstruktor** – kreira prazan raspored.
   b) **funkciju članicu print** - na standardnom izlazu ispisuje raspored predavanja u toku dana, u svakom redu po jedno predavanje u obliku: **`pocetak – predavanje`**, (npr. 08:15:00 – Matematika), počevši od prvog predavanja u toku dana.
   c) **funkciju članicu dodaj** – dodaje novo predavanje u postojeći dnevni raspored (ako termin nije zauzet).

   Za klasu Raspored treba preklopiti sljedeće operatore:
   a) **operator ()** – provjerava da li se u raspored može dodati nova aktivnost u nekom periodu, odnosno da li u rasporedu već postoji neka aktivnost sa istim terminom početka.

   Razdvojiti interfejs od implementacije, tj. funkcije članice definisati izvan definicije klase.

3. Napisati program u kojem treba:
   a) kreirati tri konstantna vremenska trenutka: 08:15:00, 10:15:00 i 09:15:00.
   b) kreirati prazan raspored pa u njega dodati tri predavanja koja počinju u 08:15:00, 10:15:00 i 09:15:00, a zatim ispisati kreirani raspored.
   c) sa standardnog ulaza učitati podatke za četvrto predavanje, pa to predavanje (ako može) dodati u kreirani raspored, i na kraju ispisati modifikovani raspored.

4. Odrediti izlaz iz sljedećeg programa:

```
#include <iostream.h>
class A
{
  public:
    A() { ++count; }
    ~A() { --count; }
    static void report() { cout << count; }
  private: static int count;
};
int A::count=7;
void f(A x) { A a; a=x; A::report(); }
main()
{
  A a, *pa, b=a;
  A::report(); f(a); A::report();
}
```

8	**9**	**7**					

Raspodjela bodova po zadacima

Zadatak	1	2	3	4	Σ
Bodovi	35%	35%	15%	15%	100%

Primjer izvršavanja programa

```
Inicijalni raspored:
  08:15:00 – Matematika
  09:15:00 – Fizika
  10:15:00 – Programiranje
Unesite termin za novo predavanje:
10 15 00
U tom terminu vec ima predavanje
Unesite termin za novo predavanje:
07 15 00
Naziv predavanja: Elektronika
Novi raspored:
  07:15:00 - Elektronika
  08:15:00 – Matematika
  09:15:00 – Fizika
  10:15:00 – Programiranje
```