

Modul RPA – Preverjanje znanja objektno orientiranega programiranja

Ime, priimek, oddelek:

Število točk: _____/36

Ocena:

Navodila za reševanje: Rešitve vpisujte v za to predvideni prostor z nalivnim peresom ali s kemičnim svinčnikom, ostali pripomočki niso dovoljeni. Število točk za posamezna vprašanja je navedeno ob nalogah. Pišite urejeno in čitljivo. Nečitljive oz. dvoumne rešitve se točkujejo z 0 točk.

1. Kdo vse lahko dostopa do zaščitenih (protected) članov razreda? (Pozor - pri tem vprašanju je možnih več odgovorov, za en napačen odgovor se naloga točkuje z 0 točkami):
- Privatne metode tako baznega kot izpeljanega razreda.
 - Vse privatne, javne in statične metode v izpeljanem razredu.
 - Do zaščitenih članov lahko dostopamo tudi v glavnem programu, vendar moramo pred tem obvezno izvesti dedovanje razredov.
 - Do teh članov lahko dostopamo tudi iz glavnega programa, če pred tem deklariramo javne vmesnike get in set v baznem razredu.
 - Virtualne in čiste virtualne metode.

____/3 točke

2. Podan je razred X in konstruktor:

```
class X { int a, b;  
    public: X (int n) {a=n; b=0;} };
```

Med katere vrste konstruktorjev bi sodil konstruktor X (int n, bool m)?

- Privzeti konstruktor
- Kopirni konstruktor
- Pretvorbeni konstruktor
- Sodi med ostale vrste konstruktorjev

____/1 točka

3. Ali moramo pri dedovanju vedno napisati besedo public pred imenom izpeljanega razreda?
- Ne to ni nujno potrebno.
 - V kolikor imamo definirane vmesnike get in set to ni potrebno, v nasprotnem primeru pa to moramo.
 - Tak je pač dogovor med programerji, podobno kot je dogovor pri imenih razredov in objektov.
 - Besedo public moramo pri dedovanju vedno napisati pred imenom izpeljanega razreda, v nasprotnem primeru gre za sintaktično napako.

____/1 točka

4. Podan je razred X, ki ima definirano metodo `void fun (char zn[], int n)`. Katere spodaj podane metode predstavljajo veljavne preobložitve metode fun? (Pozor - pri tem vprašanju je možnih več odgovorov, za en napačen odgovor se naloga točkuje z 0 točkami):

a. <code>void fun (char zn, int n)</code>	b. <code>void fun (char *zn, int n)</code>
c. <code>int fun (char string[], int m)</code>	d. <code>void fun (char &zn[], int n)</code>
e. <code>void fun (int n, char zn[])</code>	f. <code>int fun (int m, int n)=0;</code>

____/2 točki

Modul RPA – Preverjanje znanja objektno orientiranega programiranja

5. Podana je imenski prostor Kontrolka, v katerem je deklaracija razreda RPA:

```
namespace Kontrolka {  
    class RPA  
    { int dosezene_tocke;  
      string opisna_ocena;  
      static int koliko_testov;  
    public:  
      RPA()  
      {dosezene_tocke=68; opisna_ocena="dobro(3)";}   
      void izpis ()  
      {cout << dosezene_tocke << endl << opisna_ocena << endl;}  
    };  
}
```

- a) Izven razreda in izven imenskega prostora Kontrolka inicializirajte statično spremenljivko koliko_testov na vrednost 0. To storite brez uporabe rezervirane besede using namespace.

___ /2 točki

- b) Napišite vso potrebno kodo, ki bo vsakič, ko deklariramo nov objekt razreda RPA, povečala vrednost statične spremenljivke za 1 (tukaj lahko uporabite rezervirano besedo using namespace).

___ /3 točke

1. Napišite celoten program v C++, ki bo deklariral razred Matrica, ki naj vsebuje privatno 2D tabelo dimenzije 3x3. Podatkovni tip tabele naj bo definiran kot generični podatkovni tip Type. V razredu implementirajte vse preobložitve naslednjih spodaj podanih operatorjev. Začetno vrednost 2D tabeli dodeli konstruktor, tako da napolni tabelo z naključnimi vrednostmi od -50 do 50 (interval [-50..50]), razredu pa dodajte še kopirni konstruktor ter metodo za izpis vseh njegovih lastnosti.. Primeri delovanja preobloženih operatorjev, če tip Type zamenjamo s kratkimi celimi števili:

- a) Operator +

2	-4	6		-7	10	2		-5	13	6
3	8	1	+	3	-1	9	=	-1	7	14
4	5	-3		-4	5	3		2	6	0

- b) Operator *=n (primer za n=3)

2	-4	1		2	-12	1
3	8	1	* 3 =	3	24	1
4	5	-3		4	15	-3

- c) Operator +=n (primer za n=3)

Točkovnik: 36-32 odl(5), 31,5-28 pdb(4), 27,5-23 db(3), 22,5-18 zd(2), 17,5-0 nzd(1)

Modul RPA – Preverjanje znanja objektno orientiranega programiranja

2	-4	6	+ 3	=	2	-4	9
3	8	1			3	11	1
4	5	-3			7	5	0

V glavnem programu deklarirajte dva objekta tab1 in tab2 s privzetim konstruktorjem ter kličite vse njune metod in operatorje. Pri tem tip Type zamenjajte s kratkimi celimi števili (za objekt tab1) in nato še z celimi števili (int). V programu še dinamično rezervirajte kazalec na razred s pomočjo kopirnega konstruktorja, kjer tip Type zamenjate z realnimi števili. Prikažite klic operatorja +=, metode za izpis in pred zaključkom programa obvezno sprostite kazalec iz pomnilnika!

_____/21 točk (Neoptimalne rešitve bodo dobile manj točk.)

Vrst.:	Koda:
1.	
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	
13.	
14.	
15.	
16.	
17.	
18.	
19.	
20.	
21.	
22.	
23.	
24.	
25.	
26.	
27.	
28.	
29.	

Točkovnik: 36-32 odl(5), 31,5-28 pdb(4), 27,5-23 db(3), 22,5-18 zd(2), 17,5-0 nzd(1)

Modul RPA – Preverjanje znanja objektno orientiranega programiranja

30.	
31.	
32.	
33.	
34.	
35.	
36.	
37.	
38.	
39.	
40.	
41.	
42.	
43.	
44.	
45.	
46.	
47.	
48.	
49.	
50.	
51.	
52.	
53.	
54.	
55.	
56.	
57.	
58.	
59.	
60.	
61.	
62.	
63.	
64.	
65.	
66.	
67.	
68.	

Točkovnik: 36-32 odl(5), 31,5-28 pdb(4), 27,5-23 db(3), 22,5-18 zd(2), 17,5-0 nzd(1)