

Osnovi OOP – primer kolokvijuma

1. Date su klase A, B, C, D, E.

- a) Ispred pogrešne linije upisati broj greške sa spiska koja najbolje odgovara pogrešnoj liniji ili ispraviti grešku. Moguće je da u nekoj liniji postoji više od jedne greške.
Poeni: +3 za svaku otkrivenu grešku

```
public abstract class A
{
    private int a;
    private int s;
    public A(int b) { a = b; }
    public A(A a) { this.a = a; }
    public A() { }
    public int ma() { return a; }
    public virtual int af(){ return 2; }
    public static int S { get => s; }
}

public class B : A
{
    private long b;
    protected B rb;
    public B(B o) : base(o) { }
    public B(long b) : base(b) { this = b; }
    public int mb(int x) { return b*x; }
    public override int af()
    { return base.af()-2; }
    public B Rb { get => rb; }
}

public class C : B
{
    protected int c;
    public C() { }
    public C(C x) : base(c) { c = x.b; }
    public C(int x) : base() { x = c; }
    public override int af()
    { return base.af()+1; }
}

public class D : B
{
    private short d;
    public D() { rb = new C(); }
    public D(C c) : base(c.c) { rb=new C(c); }
    public D(int d) { d = d; }
    public int md() { d++; return d; }
    public override int af()
    { return -1 + base.af(); }
}

public class E : C
{
    private float e;
    public E() { e = this.mb(1); }
    public E(E e) : base(e) { e = e.a; }
    public E(B b) : base(b) { rb=new B(af()); }
    public E(float r) : base(r) { e = r; }
    public override int af()
    { return base.af()*3; }
}
```

Spisak mogućih grešaka:

1. Nedostaje **this.** zato što argument ima isto ime kao i polje
2. Polje objekta se koristi pre nego što je napravljeno
3. Ne postoji implicitna konverzija za date vrednosne tipove
4. Pogrešna dodela vrednosti između klasnih tipova
5. Dodela vrednosti između klasnih i vrednosnih tipova
6. Operator se ne može primeniti na dati tip podatka
7. Ne može se pristupiti članu zbog modifikatora pristupa
8. Pristupa se privatnom članu druge klase
9. Ne postoji odgovarajući konstruktor u osnovnoj klasi
10. Ne postoji odgovarajući konstruktor u klasi
11. Loše napisan kopirajući konstruktor

12. Implicitno se poziva podrazumevani konstruktor

osnovne klase koji nije definisan

13. Klasa nema kopirajući konstruktor
14. Beskonačna rekurzija
15. Nedostaje modifikator **abstract**
16. Pogrešan tip rezultata metoda
17. Pogrešan pristup podatku koji nije statički
18. Pogrešan pristup podatku koji je statički
19. Pogrešan pristup metodi koji nije statički
20. Pogrešan pristup metodi koji je statički
21. Pravi se objekat apstraktne klase
22. Nedefinisana promenljiva
23. Nedefinisano polje
24. Nedefinisan metod
25. Podatak treba da bude statički
26. **this** je read-only

- b) Ispred linije koda u telu metoda **main** klase **M** upisati znak "+" za ispravne linije i "-" za neispravne. Smatrati da su klase A, B, C, D, E ispravno napisane, da svi njihovi članovi imaju specifikatore pristupa i tipove koji su navedeni, i da su sve linije ispred posmatrane ispravne.
Poeni: +1 tačan odgovor, -1 netačan. 0 bez odg.

```
class M {
    public static void main(String a[]){
        B b0 = new C(1);
        C c0 = b0;
        A a0 = new E(c0);
        B b1 = new D(c0);
        B b2 = b0.rb;
        E e0 = new E(1.1);
        D d0 = new D(e0);
        D d1 = new C(e0);
        A a1 = new C(b1);
        D d2 = b1;
        D d3 = new E(d1);
        B b3 = new D(e0);
        int m = d0.mc();
        A a3 = e0.Rb;
        C c2 = new D();
        B b4 = new E(a3);
        int y = B.ms();
        C c3 = new D(e1);
        C c4 = d3;
        D d3 = c3;
        E e1 = d2;
        B b6 = new E(d1);
    }
}
```

c) Upisati vrednosti za navedene podatke:

```
A a0 = new B();
B b0 = new D();
C c0 = new E();
E e1 = new E();
A a1 = new C();
int x0 = a0.af(); // x0 =
int x1 = b0.af(); // x1 =
int x2 = c0.af(); // x2 =
int x3 = e1.af(); // x3 =
int x4 = a1.af(); // x4 =
```