

Napredno programiranje i programski jezici

14 Priprema za ispit

Fakultet tehničkih nauka, Novi Sad
23-24/Z
Dunja Vrbaški

okvirno

10-15 pitanja

30-50 minuta

negativni (prag > 3)

3 jezika

redovno, kumulativno

C++

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
#include <iostream>
using namespace std;
int main() {
    int a = 1, b, c;
    int* pa;
    int* pb;
    int* pc;

    b = ++a;
    c = b--;
    pa = &a;
    pb = &b;
    pc = &c;

    cout << ++(*pa);
    cout << (*pb)++;
    cout << --(*pc);
    return 0;
}
```

311
321
312
310
320

Uvek postoji: nijedan od ponuđenih odgovora

Može da postoji kao opcija:

- Program se neće izvršiti, dobija se poruka o gresci.
- Program će se izvršiti, ali je rezultat nepoznat.

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
#include <iostream>
using namespace std;
int main() {
    int a = 1, b, c;
    int* pa;
    int* pb;
    int* pc;

    b = ++a;
    c = b--;
    pa = &a;
    pb = &b;
    pc = &c;

    cout << ++(*pa);
    cout << (*pb)++;
    cout << --(*pc);
    return 0;
}
```

311

321

312

310

320

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
#include <iostream>
using namespace std;
class T{
    private:
        static int t;
        T(int tt){ t += tt; }
        ~T(){ --t; }
};
void printT(T t){
    cout << t.t;
}
int T::t = 1;

int main(){
    T t(2), t2(1), t3(0);
    printT(t);
    printT(t2);
    printT(t3);
    return 0;
}
```

432
444
333
321
210

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
#include <iostream>
using namespace std;
class T{
    private:
        static int t;
        T(int tt){ t += tt; }
        ~T(){ --t; }
};
void printT(T t){
    cout << t.t;
}
int T::t = 1;

int main(){
    T t(2), t2(1), t3(0);
    printT(t);
    printT(t2);
    printT(t3);
    return 0;
}
```

432
444
333
321
210

Program se neće izvršiti, dobija se poruka o gresci.

ili nijedan od ponuđenih odgovora

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
...
class A {
    public:
        virtual void f() { cout << "Otac "; }
};
class B : public A {
    public:
        void f() { cout << "Sin "; }
};
int main() {
    A a;
    B b;
    A* pa;
    B* pb;
    pb = &b;
    a.f();
    b.f();
    pa -> f();
    pb -> f();
    return 0;
}
```

```
Otac Sin Otac Sin
Otac Otac Otac Sin
Otac Otac Otac Otac
Otac Sin Sin Otac
Otac Otac Sin Otac
```

```
#include <iostream>
using namespace std;
```


(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A {  
    public:  
        virtual void f() { cout << "Otac "; }  
};  
class B : public A {  
    public:  
        void f() { cout << "Sin "; }  
};  
int main() {  
    A a;  
    B b;  
    A* pa;  
    B* pb;  
    pb = &b;  
    a.f();  
    b.f();  
    pa -> f();  
    pb -> f();  
    return 0;  
}
```

Otac Sin Otac Sin
Otac Otac Otac Sin
Otac Otac Otac Otac
Otac Sin Sin Otac
Otac Otac Sin Otac

Program će se izvršiti, ali je rezultat nepoznat.

Razmotriti različite kombinacije dodele i poziva

(C++) Sta ce biti ispisano nakon izvrsavanja sledeceg koda?

```
...  
Lekovi{ ANDOL = 543, BRUFEN, PARACETAMOL};  
  
int main() {  
    cout << PARACETAMOL << BRUFEN << ANDOL;  
    return 0;  
}
```

545544543
543544545
543543543
54301
10543

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
...  
Lekovi{ ANDOL = 543, BRUFEN, PARACETAMOL};  
int main() {  
    cout << PARACETAMOL << BRUFEN << ANDOL;  
    return 0;  
}
```

enum

545544543
543544545
543543543
54301
10543

Nije ni: PARACETAMOL, BRUFEN, ANDOL

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A{
public:
    int x, y;
    A(){x=1; y=2;}
    ~A(){cout<<x<<y;}

    friend ostream& operator<<(ostream &out,const A& a){
        out<<a.x<<a.y;
        return out;
    }
};

void fun2(A a){a.x=2; a.y=2;}
void fun1(A& a){a.x=1; a.y=1;}

int main(){
    A a;
    fun1(a);
    cout << a;
    fun2(a);
    return 0;
}
```

```
112211
111111
112222
222211
222222
```

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A{
public:
    int x, y;
    A(){x=1; y=2;}
    ~A(){cout<<x<<y;}

    friend ostream& operator<<(ostream &out,const A& a){
        out<<a.x<<a.y;
        return out;
    }
};

void fun2(A a){a.x=2; a.y=2;}
void fun1(A& a){a.x=1; a.y=1;}

int main(){
    A a;
    fun1(a);           // 1.
    cout << a;         // 2.
    fun2(a);           // 3.
    return 0;          // 4.
}
```

112211

111111

112222

222211

222222

1. fun1 → promeni sa 12 na 11. promena se dešava na samom objektu jer je prenos po referenci. nakon funkcije a je promenjen.
2. ispisuje se promenjeni a: 11
3. fun2 → Argument a se prenosi po vrednosti. Kreira se kopija. Nazovimo ga acopy. acopy ima vrednosti 11 koje se u funkciji promene na 22. Prilikom izlaska iz funkcije acopy se uništava i poziva se destruktorkoji ispisuje vrednosti za acopy: 22
4. na kraju programa uništava se i a, koji ima vrednosti 11, poziva se destruktorkoji ispisuje te vrednosti za a: 11

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
template<class X, class Z, int Y>
class GEN {
private:
    X a[Y];
    Z* d;
public:
    GEN() {
        d = new Z(2);
        int i = 0;
        do{
            a[i] = i;
            i++;
        }while(i < Y);
    }
    GEN(int p) {
        d = new Z(p - 1);
        int i = 0;
        while(i < Y){
            a[i] = p;
            i++;
        };
    }
    ~GEN() {
        for(int i = 0; i < Y; i++)
            cout << a[i];
    }
};
```

```
int main() {
    GEN g1,g2(2);
    return 0;
}
```

1220
2012
2021
2212
2122

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
template<class X, class Z, int Y>
class GEN {
private:
    X a[Y];
    Z* d;
public:
    GEN() {
        d = new Z(2);
        int i = 0;
        do{
            a[i] = i;
            i++;
        }while(i < Y);
    }
    GEN(int p) {
        d = new Z(p - 1);
        int i = 0;
        while(i < Y){
            a[i] = p;
            i++;
        };
    }
    ~GEN() {
        for(int i = 0; i < Y; i++)
            cout << a[i];
    }
};
```

1220
2012
2021
2212
2122

```
int main() {
    GEN g1,g2(2);
    return 0;
}
```

```
int main() {
    GEN<int, DinString, 5> g1,g2(2);
    return 0;
}
```

(C++) Koji iskazi su tačni?

1. Public clanu klase se moze direktno pristupati samo iz spoljasnjosti klase.
2. Kompozicija je veza izmedju klasa.
3. Destruktor je metoda koja se poziva po zelji programera.
4. Konstruktor je metoda koja kreira objekat.
5. Apstraktna klasa je klasa koja sadrzi samo apstraktne metode.

samo 1, 2 i 4
samo 1, 2, 4 i 5
samo 3, 4 i 5
samo 1 i 4
samo 2 i 4
samo 4
svi iskazi
samo 5
nijedan iskaz

*Formulacije?
Na primer: 4.*

(C++) Koji iskazi su tačni?

1. Public članu klase se može direktno pristupiti samo iz spoljnosti klase.
2. Kompozicija je veza između klasa.
3. Destruktor je metoda koja se poziva po želji programera.
4. Konstruktor je metoda koja kreira objekat.
5. Apstraktna klasa je klasa koja sadrži samo apstraktne metode.

samo 1, 2 i 4
samo 1, 2, 4 i 5
samo 3, 4 i 5
samo 1 i 4
samo 2 i 4
samo 4
svi iskazi
samo 5
nijedan iskaz

4. formulacija...

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A{
    private:
        int a, b;
    public:
        A(){a=1; b=2;}
        A(int aa=1, int bb=2){a=aa; b=bb;}
        A(const A& aa){a=aa.a; b=aa.b;}
        void ispis() {
            cout<<++a<<b++;
        }
};

int main()
{
    A a, a2(3, 4), a3(a2);
    a.ispis();
    a2.ispis();
    a3.ispis();
}
```

```
224444
133535
123434
225566
```

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A{
    private:
        int a, b;
    public:
        A(){a=1; b=2;}
        A(int aa=1, int bb=2){a=aa; b=bb;}
        A(const A& aa){a=aa.a; b=aa.b;}
        void ispis() {
            cout<<++a<<b++;
        }
};

int main()
{
    A a, a2(3, 4), a3(a2);
    a.ispis();
    a2.ispis();
    a3.ispis();
}
```

```
224444
133535
123434
225566
```

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
class A{
    public:
        virtual int oduzimanje()=0;
        virtual int sabiranje()=0;
};
class B : public A{
    private:
        int a, b;
    public:
        B(){a=2; b=++a;}
        int oduzimanje(){
            return a-b;
        }
        int mnozenje(){
            return a*b;
        }
};
int main(){
    B b;
    cout<<b.oduzimanje()<<b.mnozenje();
}
```

04
-16
6-1
40
09
90

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
class A{
public:
    virtual int oduzimanje()=0;
    virtual int sabiranje()=0;
};
class B : public A{
private:
    int a, b;
public:
    B(){a=2; b=++a;}
    int oduzimanje(){
        return a-b;
    }
    int mnozenje(){
        return a*b;
    }
};
int main(){
    B b;
    cout<<b.oduzimanje()<<b.mnozenje();
}
```

04
-16
6-1
40
09
90

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A {  
    public:  
        virtual void p() { cout<<"B"; }  
};  
class B : public A {  
    public:  
        void p() { cout<<"A"; }  
};  
  
void f(A& ra) { ra.p(); }  
  
int main() {  
    B b;  
    b.p();  
    f(b);  
    return 0;  
}
```

BA
BB
AB
AA

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A {  
    public:  
        virtual void p() { cout<<"B"; }  
};  
class B : public A {  
    public:  
        void p() { cout<<"A"; }  
};  
  
void f(A& ra) { ra.p(); }  
  
int main() {  
    B b;  
    b.p();  
    f(b);  
    return 0;  
}
```

BA
BB
AB
AA

(C++) Koji iskazi su tačni?

1. C++ program mora da ima tacno jednu main metodu.
2. U C++u postoji obrada izuzetaka (try-catch blok).
3. U C++u postoje interfejsi.
4. U C++u svaka napisana klasa mora biti u zasebnom fajlu.

samo 1, 4
samo 1, 2, 3
samo 3, 4
samo 1 i 3
samo 2 i 4
samo 1
svi iskazi

(C++) Koji iskazi su tačni?

1. C++ program mora da ima tacno jednu main metodu.
2. U C++u postoji obrada izuzetaka (try-catch blok).
3. U C++u postoje interfejsi.
4. U C++u svaka napisana klasa mora biti u zasebnom fajlu.

samo 1, 4
samo 1, 2, 3
samo 3, 4
samo 1 i 3
samo 2 i 4
samo 1
svi iskazi

1. Formulacije: metodu -> funkciju; u globalnom namespace-u...
2. da
3. Formulacije: ne postoji ključna reč i konstrukt interface
4. ne

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
void c(int &r) {  
    r = 1;  
}  
void e(int r) {  
    r = 2;  
}  
void c(int *p) {  
    *p = 3;  
}  
int main() {  
    int a = 4;  
    c(a);  
    cout<<a;  
    c(a);  
    cout<<a;  
    return 0;  
}
```

44
11
22
33
31

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
void c(int &r) {  
    r = 1;  
}  
void e(int r) {  
    r = 2;  
}  
void c(int *p) {  
    *p = 3;  
}  
int main() {  
    int a = 4;  
    c(a);  
    cout<<a;  
    c(a);  
    cout<<a;  
    return 0;  
}
```

44

11

22

33

31

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A {
public:
    A() { cout<<"AK "; }
    ~A() { cout<<"AD "; }
};
class B : public A {
public:
    B() : A() { cout<<"BK "; }
    ~B() { cout<<"BD "; }
};
class C : public A {
private:
    B b;
public:
    C() { cout<<"CK "; }
    ~C() { cout<<"CD "; }
};
int main() {
    B b;
    C c;
    return 0;
}
```

```
AK BK AK AK BK CK CD BD AD AD BD AD
AK BK AK AK BK CK CD BD AD AD AD BD
AK BK AK AK BK CK AD BD AD AD BD CD
AK BK AK AK BK CK BD AD AD AD BD CD
AK BK AK AK BK CK
AK BK CD CD BD AD
```

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A {
public:
    A() { cout<<"AK "; }
    ~A() { cout<<"AD "; }
};
class B : public A {
public:
    B() : A() { cout<<"BK "; }
    ~B() { cout<<"BD "; }
};
class C : public A {
private:
    B b;
public:
    C() { cout<<"CK "; }
    ~C() { cout<<"CD "; }
};
int main() {
    B b;
    C c;
    return 0;
}
```

AK BK AK AK BK CK CD BD AD AD BD AD
AK BK AK AK BK CK CD BD AD AD AD BD
AK BK AK AK BK CK AD BD AD AD BD CD
AK BK AK AK BK CK BD AD AD AD BD CD
AK BK AK AK BK CK
AK BK CD CD BD AD

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A{
public:
    int a;
    static int b;
    A(){a=1;}
    A(int aa){a=aa;}
    int getA() const{
        return a;
    }
    int getB() const{
        return b;
    }
};
int A::b = 2;
int main() {
    A a(3), b;
    a.a += b.b;
    a.b += b.a;
    b.a += a.b;
    b.b += a.a;
    cout<<a.a<<a.b<<b.a<<b.b;
}
```

5848
5347
5438
5887
4753

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
class A{
public:
    int a;
    static int b;
    A(){a=1;}
    A(int aa){a=aa;}
    int getA() const{
        return a;
    }
    int getB() const{
        return b;
    }
};
int A::b = 2;
int main() {
    A a(3), b;
    a.a += b.b;
    a.b += b.a;
    b.a += a.b;
    b.b += a.a;
    cout<<a.a<<a.b<<b.a<<b.b;
}
```

5848

5347

5438

5887

4753

vezbanje: cout

(C++) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
class A {
private:
    int x;
public:
    A() { x=0; }
    A(int xx) { x=xx; }
    const A& operator++() {
        x++;
        A temp(++x);
        temp.x = ++x;
        return *this;
    }
    const A operator++(int i) {
        A temp(x);
        x++;
        cout<<"B";
        return temp;
    }
};

int main() {
    A a(6);
    ++a;
    return 0;
}
```

A
B
6A
7A
6B
7B

(C++) Sta ce biti ispisano nakon izvorsavanja sledeceg koda?

```
class A {  
    private:  
        int x;  
    public:  
        A() { x=0; }  
        A(int xx) { x=xx; }  
        const A& operator++() {  
            x++;  
            A temp(++x);  
            temp.x = ++x;  
            return *this;  
        }  
        const A operator++(int i) {  
            A temp(x);  
            x++;  
            cout<<"B";  
            return temp;  
        }  
};  
int main() {  
    A a(6);  
    ++a;  
    return 0;  
}
```

A
B
6A
7A
6B
7B

nema ispisa "A"
Izvršiće se, ali neće ništa ispisati

Java

(Java) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
public class A {  
    public int a;  
}  
  
public class Test {  
    public static void main(String[] args){  
        A a1 = new A();  
        A a2 = new A();  
        A a3 = new A();  
        A a4 = new A();  
  
        a1.a = 1; a2.a = 2;  
        a3.a = 3; a4.a = 4;  
  
        ArrayList<A> lista;  
        a3 = a2;  
        lista.add(a1); lista.add(a2);  
        lista.add(a3); lista.add(a4);  
        a4 = a1;  
        for (int i = 0; i < 4; i++) {  
            System.out.print(lista.get(i).a);  
        }  
    }  
}
```

1224
1234
1324
1222
1221

(Java) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
public class A {  
    public int a;  
}  
  
public class Test {  
    public static void main(String[] args){  
        A a1 = new A();  
        A a2 = new A();  
        A a3 = new A();  
        A a4 = new A();  
  
        a1.a = 1; a2.a = 2;  
        a3.a = 3; a4.a = 4;  
  
        ArrayList<A> lista;  
        a3 = a2;  
        lista.add(a1); lista.add(a2);  
        lista.add(a3); lista.add(a4);  
        a4 = a1;  
        for (int i = 0; i < 4; i++) {  
            System.out.print(lista.get(i).a);  
        }  
    }  
}
```

1224
1234
1324
1222
1221

= new ArrayList<A>();

(Java) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
public class Test{  
    public static void main(String[] args) {  
        String s = "Pera;;Ana;;Jovan";  
        String parts[] = s.split(";");  
        System.out.println(parts.length);  
    }  
}
```

5
7
4
3
16

(Java) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
public class Test{  
    public static void main(String[] args) {  
        String s = "Pera;;Ana;;Jovan";  
        String parts[] = s.split(";");  
        System.out.println(parts.length);  
    }  
}
```

5

7

4

3

16

(Java) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
public class A {  
    public String s;  
    public int a;  
    public A(){  
        s = "PetarPetrovic";  
        a = 0;  
    }  
    public String getUnifiedCase(){  
        if(a == 0)  
            return s.toUpperCase();  
        return s.toLowerCase();  
    }  
    @Override  
    public String toString() {  
        return getUnifiedCase() + a;  
    }  
}  
  
public class Test {  
    public static void main(String[] args){  
        A a = new A();  
        System.out.print(a);  
    }  
}
```

```
PETARPETROVIC+0  
PetarPetrovic+0  
petarpetrovic+0  
PETARPETROVIC0  
PetarPetrovic0  
petarpetrovic0  
PETARPETROVIC  
petarpetrovic
```

(Java) Sta ce biti ispisano nakon izvršavanja sledeceg koda?

```
public class A {  
    public String s;  
    public int a;  
    public A(){  
        s = "PetarPetrovic";  
        a = 0;  
    }  
    public String getUnifiedCase(){  
        if(a == 0)  
            return s.toUpperCase();  
        return s.toLowerCase();  
    }  
    @Override  
    public String toString() {  
        return getUnifiedCase() + a;  
    }  
}  
  
public class Test {  
    public static void main(String[] args){  
        A a = new A();  
        System.out.print(a);  
    }  
}
```

PETARPETROVIC+0
PetarPetrovic+0
petarpetrovic+0
PETARPETROVIC0
PetarPetrovic0
petarpetrovic0
PETARPETROVIC
petarpetrovic

Python

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
a = 0
while a < 5:
    print(a)
    a += 1
```

```
a = 0
while a < 5:
    print(a)
a += 1
```

```
while a < 5:
    print(a)
    a += 1
```

-01234

-12345

-012345

-1234

-javlja se problem sa nedostatkom memorije
(program se "beskonačno dugo izvršava")

-program se neće izvršiti, jer ćemo dobiti upozorenje o
gresci.

ignorirati nove redove prilikom ispisa

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
for i in range(5):  
    print(i)
```

```
for i in range(1, 5):  
    print(i)
```

-01234
-12345
-012345
-1234
-javlja se problem sa nedostatkom memorije
(program se "beskonačno dugo izvršava")
-program se neće izvršiti, jer ćemo dobiti upozorenje o
gresci.

ignorirati nove redove prilikom ispisa

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
student = ("Petar", "Petrovic", 123456)
bri = (42, 2020)
student += bri
student[3] = 45
print(student)
```

- ('Petar', 'Petrovic', 123456)
- ('Petar', 'Petrovic', 123456, 42, 2020)
- ('Petar', 'Petrovic', 123456, 45, 2020)
- program se neće izvršiti, jer ćemo dobiti upozorenje o gresci.

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
student = ("Petar", "Petrovic", 123456)
bri = (42, 2020)
student += bri
student[3] = 45
print(student)
```

- ('Petar', 'Petrovic', 123456)
- ('Petar', 'Petrovic', 123456, 42, 2020)
- ('Petar', 'Petrovic', 123456, 45, 2020)
- program se neće izvršiti, jer ćemo dobiti upozorenje o gresci.

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
t = (3, 5)
print(t)

(x, y) = (3, 5)
print(x, y)
```

```
-(3, 5)      3 5
-3, 5        3, 5
-(3, 5)      (3, 5)
-(3, 5)      3, 5
-3, 5        3 5
```

-program se neće izvršiti, jer ćemo dobiti upozorenje o grešci.

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
t = (3, 5)
print(t)

(x, y) = (3, 5)
print(x, y)
```

-(3, 5)	3 5
-3, 5	3, 5
-(3, 5)	(3, 5)
-(3, 5)	3, 5
-3, 5	3 5

-program se neće izvršiti, jer ćemo dobiti upozorenje o grešci.

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
data = [1, 15, -2, 7, 8]
data1 = [num ** 2 for num in data]
print(data1)
```

-[1, 225, 4, 49, 64]
-[2, 30, -4, 14, 16]
-program se neće izvršiti, jer ćemo dobiti upozorenje o gresci.

```
data = [i for i in range(5)]
print(data)
```

-[0,1,2,3,4]
-[0,1,2,3,5]
-[1,2,3,4]
-[1,2,3,4,5]
-program se neće izvršiti, jer ćemo dobiti upozorenje o gresci.

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
class Pravougaonik:

    def __init__(self, a = 1, b = 2):
        self.a = a
        self.b = b

    def __str__(self):
        return f"a = {self.a}, b = {self.b}"

p1 = Pravougaonik()
print(p1)

p2 = Pravougaonik(3, 5)
print(p2)

p3 = Pravougaonik(b = 3, a = 5)
print(p3)
```

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
class Pravougaonik:
    br_instanci = 0

    def __init__(self, a = 1, b = 2):
        self.a = a
        self.b = b
        Pravougaonik.br_instanci += 1

    def __str__(self):
        return f"a = {self.a}, b = {self.b}"

p1 = Pravougaonik()
print(p1)

p2 = Pravougaonik(3, 5)
print(p2)

p3 = Pravougaonik(b = 3, a = 5)
print(p3)

print(Pravougaonik.br_instanci)
```

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
class A:
    A(int x, int y):
        self.x = x;
        self.y = y;

A a1 = A(3,4)
A a2 = A(5,6)
A a3 = A(7,8)
A a4 = A(9,10)

a3.x = 5;
a4.y = 8;

lista = []
lista.append(a3);
lista.append(a1);

a3 = a4;
a2 = a1;

lista.append(a2);
lista.append(a4);

for (A a : lista.values())
    print(a.x + " " + a.y)
```

-58343498
-58355698
-783434910
+Program se neće izvršiti, jer ćemo dobiti upozorenje o gresci.

ignorirati nove redove prilikom ispisa

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
class A:
    A(int x, int y):
        self.x = x
        self.y = y

A a1 = A(3,4)
A a2 = A(5,6)
A a3 = A(7,8)
A a4 = A(9,10)

a3.x = 5
a4.y = 8

lista = []
lista.append(a3)
lista.append(a1)

a3 = a4
a2 = a1

lista.append(a2)
lista.append(a4)

for (A a : lista.values())
    print(a.x + " " + a.y)
```

-58343498
-58355698
-783434910
+Program se neće izvršiti, jer ćemo dobiti upozorenje o grešci.

nije ovo sve...

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
class A:
    A(int x, int y):
        self.x = x
        self.y = y

A a1 = A(3,4)
A a2 = A(5,6)
A a3 = A(7,8)
A a4 = A(9,10)

a3.x = 5
a4.y = 8

lista = []
lista.append(a3)
lista.append(a1)

a3 = a4
a2 = a1

lista.append(a2)
lista.append(a4)

for (A a : lista.values())
    print(a.x + " " + a.y)
```

-58343498
-58355698
-783434910
+Program se neće izvršiti, jer ćemo dobiti upozorenje o grešci.

Šta će biti ispisano nakon izvršavanja sledećeg koda?

```
class A:
    def __init__(self, x, y):
        self.x = x
        self.y = y

a1 = A(3,4)
a2 = A(5,6)
a3 = A(7,8)
a4 = A(9,10)

a3.x = 5
a4.y = 8

lista = []
lista.append(a3)
lista.append(a1)

a3 = a4
a2 = a1

lista.append(a2)
lista.append(a4)

for a in lista:
    print(f"{a.x} {a.y}")
```

-58343498

-58355698

-783434910

+Program se neće izvršiti, jer ćemo dobiti upozorenje o gresci.

ignorirati nove redove prilikom ispisa

- U obzir dolaze i drugačija pitanja
- Sve što je rađeno na predavanjima

Java/C++

- dodela, upoređivanje (equals)
- konstruktori, inicijalizacija, prava pristupa
- nasleđivanje, virtualnost
- apstraktne klase
- nizovi (lista, mapa)
- iskazi
- static
- izuzeci

C++

- reference
- prenos parametara - vrednost, referenca, adresa
- const
- preklapanje operatora
- generičke - Implementacija

Java

- stringovi
- generičke - ArrayList, HashMap
- interfejs

Java vs C++

Python

- funkcije višeg reda
- tuple, list, dictionary - promenljivi ili ne, operatori, funkcije
- konstruktor, str, public
- polja klase
- nasleđivanje (klase A, B, C)
- ...

Materijal sa vežbi

Java vs C++ vs Python

Zadaci iz C++/Java → Python
probati ako ima smisla

ZADATAK

Isprobati sve primere. Pre otkrivanja narednog slajda sa rešenjem, sami pretpostavite odgovor pa tek onda proverite.

Svuda gde postoji greška, ispraviti je i pokušati rešavanje. Prvo smisliti odgovor, a onda pokrenuti program i porveriti.