



FER
Preddiplomski studij

Računarstvo

Telekomunikacije i
informatika

Ime i prezime: _____

Mat. br. _____

Grupa B

Informacija, logika i jezici

1. MEĐUISPIT, 29. travanj 2013.

Izjavljujem da tijekom izrade ovog međuispita neću od drugoga primiti niti drugome pružiti pomoć te da se neću koristiti nedopuštenim sredstvima. Ove su radnje teška povreda Kodeksa ponašanja te mogu uzrokovati i trajno isključenje s Fakulteta.

Također izjavljujem da mi zdravstveno stanje dopušta pisanje ovog međuispita.

Ak. g. 2012./2013.

Vlastoručni potpis: _____

Sva pitanja u međuispitu koja sadrže programski kôd ili su na bilo koji način povezana sa programskim kôdom odnose se isključivo na programski jezik Javu!

Zadatak 1. (1 bod) Kojoj skupini osnovnih programskih paradigma pripada Java?

- A. Java pripada imperativnoj skupini osnovnih programskih paradigma.
- B. Java pripada funkcionalnoj skupini osnovnih programskih paradigma.
- C. Java pripada objektno orijentiranoj skupini osnovnih programskih paradigma.
- D. Java pripada logičkoj skupini osnovnih programskih paradigma.

Zadatak 2. (1 bod) U Javi se mogu koristiti apstraktne metode i klase. Koje od navedenih tvrdnji o apstraktnim metodama/klasama su točne?

- (1) Apstraktna klasa se ne može instancirati.
 - (2) Konstruktori ne mogu biti apstraktni.
 - (3) Potklasa apstraktne klase mora deklarirati apstraktne metode.
 - (4) Statičke metode mogu biti deklarirane kao apstraktne.
- A. Točne su tvrdnje (1), (2) i (3).
 - B. Točna je samo tvrdnja (1).
 - C. Točne su tvrdnje (1) i (2).
 - D. Točna je samo tvrdnja (2).

Zadatak 3. (1 bod) U Javi su klase organizirane u pakete. Koje su klase i paketi dostupni nakon prevođenja sljedeće naredbe? `import hr.fer.tel.mojpaket.*`

- A. Dostupni su samo pod-paketi paketa `hr.fer.tel.mojpaket`.
- B. Dostupni su svi pod-paketi i klase paketa `hr.fer.tel.mojpaket`.
- C. Dostupni su svi pod-paketi paketa `hr.fer.tel.mojpaket`, ali ne i njegove klase.
- D. Dostupne su samo klase koje se nalaze u paketu `hr.fer.tel.mojpaket`.

Zadatak 4. (1 bod) U Javi se uz klase mogu koristiti i sučelja. Što vrijedi za sučelje (engl. *Interface*)?

- A. Sve metode koje se nalaze u njemu potrebno je implementirati.
- B. Metode je potrebno samo deklarirati.
- C. Sučelje nije moguće koristiti kao vrstu podataka.
- D. U sučelju su sve metode privatne.

Zadatak 5. (1 bod) U Javi se koriste različiti tipovi podataka. Ako su definirane varijable `s1` i `s2`, koji od navedenih izraza (1) – (5) neće izazvati pogrešku? `String s1 = "a"; String s2 = "b";`

- | | |
|--|--|
| <ul style="list-style-type: none"> A. Izrazi (1), (2) i (3). B. Izrazi (1), (3) i (5). C. Izrazi (1), (4) i (5). D. Izrazi (3), (4) i (5). | <ul style="list-style-type: none"> (1) <code>String s3 = s1 + s2;</code> (2) <code>String s3 = s1 - s2;</code> (3) <code>s1 <= s2;</code> (4) <code>s1.compareTo(s2);</code> (5) <code>int m = s1.length();</code> |
|--|--|

Zadatak 6. (1 bod) Koja dva osnovna sučelja sačinjavaju kolekcije u Javi?

- | | |
|---|--|
| <ul style="list-style-type: none"> A. <code>List</code>, <code>Iterator</code> C. <code>List</code>, <code>Map</code> | <ul style="list-style-type: none"> B. <code>Map</code>, <code>Comparator</code> D. <code>List</code>, <code>Set</code> |
|---|--|

Zadatak 7. (1 bod) Što će se dogoditi kod prevodenja i izvođenja sljedećeg kôda u metodi main()?

```
String s = new String("Computer");
if(s == "Computer") System.out.println("Equal A");
if(s.equals("Computer")) System.out.println("Equal B");
```

- A. Pojavit će se pogreška kod prevodenja jer klasa String ne podržava operator ==.
- B. Ispisat će se samo "Equal A".
- C. Ispisat će se samo "Equal B".
- D. Ispisat će se i "Equal A" i "Equal B".

Zadatak 8. (1 bod) Prilikom korištenja različitih tipova podataka valja biti oprezan pri pridjeljivanju vrijednosti. Koji je rezultat izvođenja sljedećeg programskog odsječka?

```
byte b;
int a = 10;
b = a;
System.out.println("Vrijednost b je " + b);
```

- A. Vrijednost b je 10.
- B. Ništa se neće ispisati jer će prevoditelj javiti pogrešku.
- C. Dolazi do bacanja iznimke ClassCastException.
- D. Vrijednost b je [Ljava.lang.Byte;@1037c71.

Zadatak 9. (2 boda) Klase u Javi se instanciraju koristeći konstruktore. Je li sljedeći kôdinstanciranja ispravan?

```
// Datoteka: Point.java
public class Point {
    double x; // x - koordinata točke u ravnini
    double y; // y - koordinata točke u ravnini

    public Point(double x, double y) {
        this.x=x;
        this.y=y;
    }
}
// Datoteka: Test.java
public class Test{
    public static void main(String[] args){
        Point point = new Point ();
    }
}
```

- A. Pogrešno je što podrazumijevani konstruktor u obje klase nije definiran, a mora biti definiran.
- B. Nema pogreške jer uvijek postoji podrazumijevani konstruktor bez parametara.
- C. Pogrešno je pozivati konstruktor koji u ovom slučaju nije eksplicitno definiran.
- D. Nema pogreške jer se konstruktor nasljeđuje iz klase Object.

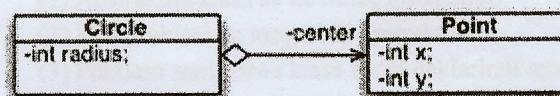
Zadatak 10. (1 bod) Koje su dvije osnovne klase koje predstavljaju pristup tokovima podataka oktetima (byte)?

- | | |
|------------------------------|---------------------------|
| A. InputReader, OutputReader | B. FileReader, FileWriter |
| C. InputStream, OutputStream | D. Reader, Writer |

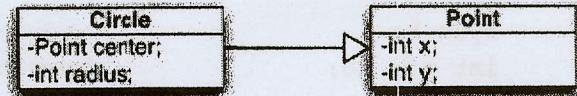
Zadatak 11. (2 boda) Zadane su klase Circle i Point. Koji od navedenih UML-ovih dijagrama ispravno prikazuje navedenu ovisnost?

```
//Datoteka Circle.java.
public class Circle {
    private Point center;
    private int radius;
    ...
    public Circle() {
        center = new Point();
        radius = 10;
    }
}
```

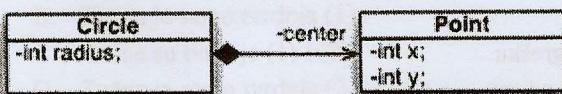
```
//Datoteka Point.java.
public class Point {
    private int x,y;
    ...
}
```



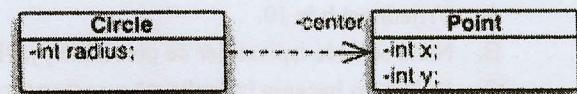
A.



B.



C.



D.

Zadatak 12. (2 boda) Što će se dogoditi prilikom prevodenja/pokretanja sljedećeg programskega odsječka koji se nalazi unutar istog paketa?

```
// Datoteka: ClassA.java
1. public class ClassA {
2.     private ClassA() {}
3. }
```

```
// Datoteka: ClassB.java
1. public class ClassB {
2.     public ClassB() {
3.         ClassA klasa = new ClassA();
4.     }
5. }
```

```
// Datoteka: Main.java
1. public class Main {
2.     public static void main(String[] args) {
3.         ClassB klasa = new ClassB();
4.     }
5. }
```

- A. Javit će se pogreška prilikom pokretanja u 2. retku datoteke ClassA.java.
- B. Javit će se pogreška prilikom prevodenja u 2. retku datoteke ClassA.java.
- C. Javit će se pogreška prilikom prevodenja u 3. retku datoteke ClassB.java.
- D. Programski odsječak će se izvršiti bez pogrešaka.

Zadatak 13. (1 bod) Kako izgleda deklaracija XML-ovog dokumenta?

- A. <?xml version="1.0"?>
- B. <?xml verion="1.0"->
- C. <!Content-Type: text/xml>
- D. <!DOCTYPE format="xml" version="1.0">

Izvorni kôd 1. Klasa EquilateralPolygon predstavlja jednakostranične poligone i sadržava njihovu zajedničku karakteristiku, duljinu stranice. Klase EquilateralTriangle i Square predstavljaju jednakostraničan trokut i kvadrat, posebne slučajevje jednakostraničnog poligona kojeg nasleđuju. Ovaj izvorni kôd sadrži dijelove programa koji se ispituju u zadacima 14, 15, 16 i 17.

```

public class EquilateralPolygon {
    private int sideLength;

    public String toString() {
        return "Equilateral:" + getSideLength();
    }

    public int getSideLength() {
        return sideLength;
    }

    public void setSideLength(int sideLength) {
        if (sideLength < 0)
            throw new IllegalArgumentException("Side length must be non-negative!");
        this.sideLength = sideLength;
    }

    public class EquilateralTriangle extends EquilateralPolygon {
        private double altitude;

        public EquilateralTriangle(int sideLength) {
            setSideLength(sideLength);
        }

        public String toString() {
            return "EquilateralTriangle:" + getSideLength();
        }

        public double getAltitude() {
            return altitude;
        }

        public void setSideLength(int sideLength) {
            super.setSideLength(sideLength);
            altitude = Math.sqrt(3) / 2 * getSideLength();
        }
    }

    public class Square extends EquilateralPolygon {
        private double diagonalLength;

        public Square(int sideLength) {
            setSideLength(sideLength);
        }

        public String toString() {
            return "Square:" + getSideLength();
        }

        public void setSideLength(int sideLength) {
            super.setSideLength(sideLength);
            diagonalLength = Math.sqrt(2) * getSideLength();
        }
    }
}

// Nastavak klase Square iz lijevog stupca...
public double getDiagonalLength() {
    return diagonalLength;
}

public class Main {
    private _____ list;

    public Main() {
        list = new _____();
        list.add(new Square(2));
        list.add(new EquilateralTriangle(3));
        list.add(new Square(4));
        list.add(new EquilateralTriangle(5));
        list.add(new Square(6));
        list.add(new EquilateralTriangle(7));
    }

    public void removeSmallerThan(int lengthLowerBound) {
        _____
    }

    public static void printout() {
        EquilateralPolygon p = new EquilateralPolygon();
        EquilateralTriangle t = new EquilateralTriangle(3);
        EquilateralPolygon ps = new Square(2);
        System.out.print(" " + p + " " + t + " " + ps);
    }

    public static void assignment() {
        boolean b1 = false;
        int i = -1;
        double d = 10.1;

        EquilateralTriangle t = new EquilateralTriangle(1);
        EquilateralPolygon p = new EquilateralPolygon();
        // ubaciti kod umjesto komentara
    }
}

```

Zadatak 14. (2 boda) Ako u Izvornom kôdu 1 umjesto komentara u metodi assignment (dno desnog stupca) klase Main umetnemo jedan od ponuđenih odgovora, koji će se od njih prevesti i izvesti bez pogreške?

A. p = t;

B. t = p;

C. b1 = i;

D. i = d;

Zadatak 15. (2 boda) Ako u Izvornom kôdu 1 pozovemo metodu Main.printout() na standardnom će se izlazu ispisati:

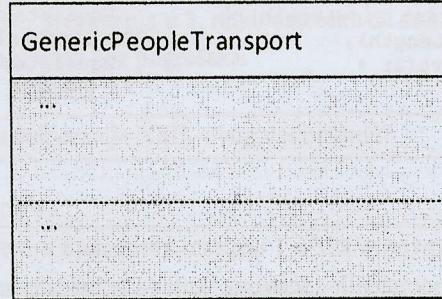
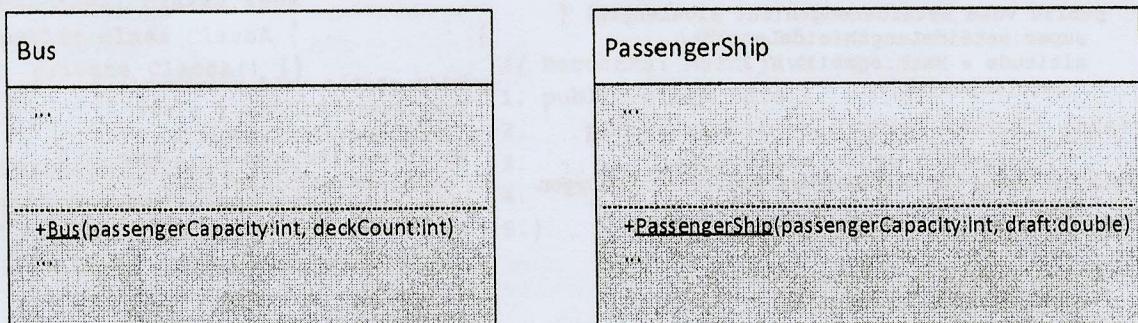
- A. Equilateral:0 EquilateralTriangle:3 Equilateral:0
- B. Equilateral:0 EquilateralTriangle:3 Square:0
- C. Equilateral:0 EquilateralTriangle:3 Square:2
- D. Equilateral:0 EquilateralTriangle:3 Equilateral:2

Zadatak 16. (2 boda) Uzimajući u obzir parametrizirane klase tj. genericse, nadopunite deklaraciju atributa list klase Main i pridruživanje nove liste atributu u konstruktoru iste klase na predviđenim crtama u Izvornom kôdu 1. Prilikom same deklaracije atributa nemojte ograničiti kasnije korištenje na točno određenu implementaciju liste.

Zadatak 17. (2 boda) Dopišite realizaciju metode removeSmallerThan klase Main u Izvorni kôd 1. Metoda mora iz liste list izbaciti sve jednakostanične poligone čija je stranica manja od parametra lengthLowerBound.

Opis 1. U agenciji koja se bavi prijevozom putnika, potrebno je ostvariti programsko rješenje koje objedinjuje sve podržane modalitete prijevoza (autobus i putnički brod). Prikazan je izvorni kôd dijela jednog takvog rješenja. Klasa Bus ima atribute: deckCount (broj katova) i passengerCapacity (kapacitet putnika). Klasa PassengerShip ima atribute: draft (dubina gaza) i passengerCapacity (kapacitet putnika). Koristeći ovaj opis riješite zadatke 18 i 19.

Zadatak 18. (2 boda) U skladu s Opisom 1 nadopunite dijagrame klasa upisivanjem atributa te odnosa među klasama na odgovarajuća mjesta. Rješenje mora biti u skladu s objektno orijentiranom paradigmom. Dijagrami klasa moraju sadržavati oznake modifikatora pristupa, vrste ulaznih i izlaznih parametara i pripadnost metode klasi ili instanci.



Zadatak 19. (4 boda) U skladu s **Opisom 1** nadopunite kôd upisivanjem atributa i odnosa među klasama na za to predviđena mjesta u odgovarajućim klasama, te dopisivanjem *get* i *set* metoda ispod mjesta u klasama označenih sa "...", pazeći da postojeći kod bude ispravan. Rješenje mora biti u skladu s objektno orijentiranim paradigmom.

```
public class Bus
```

```
{
```

```
public Bus(int passengerCapacity,
           int deckCount) {
    setPassengerCapacity(
        passengerCapacity);
    setDeckCount(deckCount);
}
```

```
...
```

```
}
```

```
public class GenericPeopleTransport
```

```
...
```

```
}
```

```
public class PassengerShip
```

```
{
```

```
public PassengerShip(int
                      passengerCapacity, double draft) {
    setPassengerCapacity(
        passengerCapacity);
    setDraft(draft);
}
```

```
...
```

```
}
```

```
{
```