Septembar 2020.
Kvalifikacioni test

K1. Šta će biti rezultat izvršavanja narednog koda ukoliko nema kompajlerskih ili grešaka u izvršavanju? Bilo koji scenario da je u pitanju, obrazložiti. Ukoliko postoji problem dati predlog za njegovo rešavanje.

```
public class InitialTest {
   static int x = 5;
   InitialTest() {
            x++;
   }
   public static void main(String[] args) {
       InitialTest a = new InitialTest();
       InitialTest b = new InitialTest();
       InitialTest c;
       b = a;
       a.x++;
       System.out.println(c.x);
    }
}
```

K2. Šta su članovi objekata, a šta članovi klasa?

K3. Šta će biti rezultat izvšavanja sledećeg koda? *Obrazložiti detaljno.*

```
class AnimalCreator {
     public static void main(String [] args) {
            Animal a = new Animal();
            Cow c = (Cow) a;
            System.out.println(c.name);

     }
     }
     class Animal {
           public String name = "Zivotinja";
     }
     class Cow extends Animal {
           String name = "Krava";
     }
```

K4. Neka je dat sledeći kod:

```
1. public class Test{
2.     public static void main (String[] args) {
3.           B b = new B();
4.           b.jaSam();
5.     }
6.}
7. class A {
8.     A(int x) { System.out.println("A:"+ x);}
9. }
10. class B extends A {
11.     public void jaSam() {
12.           System.out.println("Ja sam B");
13.     }}
```

Šta će se desiti pri njegovom pokretanju? (Obrazložiti odgovor)

K5. Šta je JVM, a šta JRE?

September 2020.
Završni deo ispita – TEST

- Pitanja A kategorije su osnovna i bez davanja tačnih odgovora na njih nije moguće položiti završni deo ispita.
- Na pitanja B kategorije nije obavezno dati odgovor da bi ostala pitanja bila bodovana.

kategorija A

A1. super i super() (Objasniti I dati primer)

A2. Overloading.

A3. Paketi.

A4. Statičko li dinamičko vezivanje? Na šta se odnosi termin vezivanje? Kada se koje primenjuje?

A5. Na šta se može primeniti ključna reč synchronized? Šta je efekat?

kategorija B

B1. Šta će biti rezultat izvršavanja sledećeg koda? Obrazložiti.

```
public class Client {
 static void doCalc(byte... a) {
 System.out.print("byte...");
 }
 static void doCalc(long a, long b) {
 System.out.print("long, long");
 }
 static void doCalc(Byte s1, Byte s2) {
 System.out.print("Byte, Byte");
 }
 public static void main (String[] args) {
 byte b = 5;
 doCalc(b, b);
 }
}
```

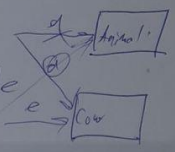
A. byte…
B. long, long
C. Byte, Byte
D. compilation error

B2. Da li dati kod ima kompajlerskih ili grešaka u izvršavanju. Ako nema šta je rezultat njegovog izvršavanja? Ako ima, kako ih rešiti?

```
public class Test {
      int variable = 1;
      Test() {
            variable++;
      }
      public double GetVariable2(float x) {
            return variable + x;
      }
      public double GetVariable2(double x) {
            return variable * x;
      }
      public static void main(String [] args) {
            Test t = new Test();
            System.out.println(GetVariable2(12.3));
      }
}
```

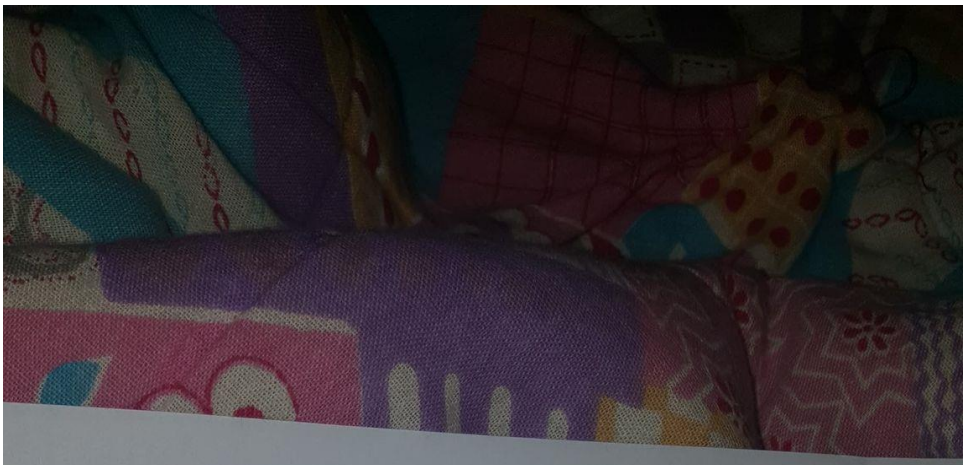
B3. Šta će biti rezultat izvršavanja sledećeg koda? Obrazložiti odgovor.

```
class AnimalCreator {
public static void main(String [] args) {
                  Animal a = new Animal();
                  Cow c = new Cow();
                  a = c;
                  a.getAnimal();
                  c = (Cow) a;
                  c.getAnimal();
            }
}
class Animal {
      static Animal getAnimal() {
            System.out.println("Animal");
      }
}
class Cow extends Animal {
      static Animal getAnimal() {
            System.out.println("Cow");
      }
}
```



B4. *Neka su definisana dva tipa niti, od kojih jedna koristi objekat skladišta umanjujući njegovo stanje za 1, a druga dodeljuje uvećava stanje skladišta za 1.*

```
class Proizvodjac extends Thread {
      private Skladiste mojeskladiste;
      private int broj;
      private int k;
public Proizvodjac(Skladiste sk, int broj) {
            mojeskladiste = sk;
            this.broj = broj;
            }
public void run() {
      for (int i = 0; i < 10; i++) { mojeskladiste.dodaj(); }
}
```




```
}
class Korisnik extends Thread {
        private Skladiste mojeskladiste;
        private int broj;

 public Korisnik(Skladiste sk, int broj) {
         mojeskladiste = sk;
         this.broj = broj;
   }
 public void run() {
         int procitani_broj = 0;
         for (int i = 0; i < 10; i++)  procitani_broj =  mojeskladiste.uzmi();
    }
 }
```

Definisati klasu Skladiste tako da bude thread-safe, onemogući istovremeni pristup/ažuriranje stanja skladišta i obezbedi da nema čitanja dok skladišta nije puno, niti pisanja ako skladište nije prezno. Napisati telo main metoda testne klase u kom se instanciraju po dve niti korisnika i potrošača i startuju.