Objektovo-orientované programovanie **B**
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Test – 4. apríl 2017

Priezvisko:	tlačeným písmom
Meno:	

1b	
2b	

1	
2	
3	
4	
5	
6	
7	
8	
9	

Test trvá 30 minút.

Odpovede na otázky vpíšte do tabuľky. Hodnotia sa len odpovede v tabuľke. Pri otázkach s výberom odpovede je len jedna možnosť správna.

V prípade opravy jasne vyznačte odpoveď, ktorá platí. Každá správna odpoveď má hodnotu vyznačenú v otázke. Nesprávna, nejednoznačná alebo neúplná odpoveď má hodnotu 0 bodov. Postup riešenia sa nehodnotí.

Poškodený list nebude uznaný.

1. (2 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
abstract class P {
   void m(N d) {
      System.out.print("p");
   }
}

interface N {
   void op(A e);
   void op(B e);
}

class A extends P {
   public void m(N d) {
      super.m(d);
      d.op(this);
   }
}

class B extends P {
   public void m(N d) {
      super.m(d);
      d.op(this);
   }
}

class X implements N {
   public void op(A e) {
      System.out.print("ax ");
   }
   public void op(B e) {
      System.out.print("bx ");
   }
}

class Y implements N {
   public void op(A e) {
      System.out.print("ay ");
   }
   public void op(B e) {
      System.out.print("by ");
   }
}

public class M {
   public static void main(String[] args) {
      P o1 = new A();
      A o2 = new A();
      B o3 = new B();
      P o4 = new B();

      o1.m(new Y());
      o2.m(new X());
      o3.m(new X());
      o4.m(new Y());
   }
}
```

2. (1 b) Aká je hlavná výhoda použitia anonymných tried?

(a) šetrenie pamäťou
(b) zvýšenie bezpečnosti
(c) lepšie vyjadrenie zámeru
(d) zmenšenie programu
(e) zrýchlenie programu

3. (1 b) Daná je trieda

```
public class C implements Serializable {
   public String id;
   private List<C> l = new ArrayList<>();

   public C(String id) {
       this.id = id;
   }
   public void m(C... c) {
       for (C e : c)
           l.add(e);
   }
}
```

Dané sú inštancie triedy `C`:

```
C a = new C("a");
C b = new C("b");
C c = new C("c");
C d = new C("d");
b.m(a, d);
c.m(a, b, c, d);
d.m(b);
```

Pre serializáciu týchto inštancií platí, že

(a) najviac inštancií sa korektne serializuje, ak sa za začiatok serializácie zvolí objekt `d`
(b) najviac inštancií sa korektne serializuje, ak sa za začiatok serializácie zvolí objekt `c`
(c) je jedno, ktorý objekt sa zvolí za začiatok serializácie, lebo **private** zabráni serializácii atribútu `l`
(d) najviac inštancií sa korektne serializuje, ak sa za začiatok serializácie zvolí objekt `b`
(e) najviac inštancií sa korektne serializuje, ak sa za začiatok serializácie zvolí objekt `a`

4. (1 b) Zabezpečenie vykonania určitej akcie po kliknutí na tlačidlo sa v rámcoch na zabezpečenie grafického používateľského rozhrania, akými sú JavaFX a Swing, sa dosahuje

(a) sledovaním, či je tlačidlo kliknuté v nekonečnej slučke vo vlastnej niti a následným vykonaním akcie
(b) sledovaním, či je tlačidlo kliknuté v nekonečnej slučke v niti na odosielanie/vykonanie udalostí (event dispatching thread) a následným vykonaním akcie
(c) pridaním tejto akcie vo forme prijímača/spracovateľa (listener/handler) udalosti kliknutia registrovaného pre dané tlačidlo
(d) synchronizáciou udalosti v hlavnej metóde programu
(e) špeciálnou jazykovou konštrukciou na zachytenie udalosti kliknutia a následným vykonaním akcie, ak sa zachytená udalosť týka príslušného tlačidla

5. (1 b) Daný je nasledujúci kód v Jave:

```
while (getObject(o)) {
   if (o instanceof A)
      ((A) o).opa();
   else if (o instanceof B)
      ((B) o).opb();
   else
      ;
}
```

Tento kód porušuje princíp otvorenosti a uzavretosti, lebo

(a) triedy a objekty sú na rozdielnej úrovni abstrakcie a netreba ich porovnávať
(b) pribúdaním ďalších tried, ktoré treba zobrať do úvahy, slučka **while** sa stane neprehľadnou
(c) používa operátor **instanceof** namiesto metódy `isInstance`()
(d) nedefinovaním akcie pre **else** nie sú uzavreté všetky možnosti
(e) ak pribudne ďalšia trieda, ktorú treba zobrať do úvahy, treba zmeniť slučku **while**

6. (1 b) Opätovné využitie štruktúry a správania triedy v inej triede s možnosťou rozšírenia štruktúry a zmeny správania sa v objektovo-orientovanom programovaní označuje ako

(a) dedenie
(b) generickosť
(c) segregácia
(d) agregácia
(e) integrácia

7. (1 b) Zachytenie výnimky v Jave

(a) signalizuje výnimočnú situáciu klientskemu kódu
(b) ohrozuje integritu programu
(c) automaticky opravuje vzniknutú chybu
(d) umožňuje jej spracovanie a pokračovanie v programe
(e) automaticky zastavuje program

8. (1 b) Synchronizovať nestatickú metódu v Jave znamená zabezpečiť

(a) rychlejšie vykonávanie tejto metódy
(b) rovnomerné prepínanie nití pri vykonávaní tejto metódy
(c) zamknutie objektu triedy (**class**) kým sa táto metóda nevykoná
(d) nemennosť tejto metódy v odvodených triedach
(e) zamknutie aktuálneho objektu (**this**) kým sa táto metóda nevykoná

9. (1 b) Ktorý návrhový vzor implementuje tento kód v Jave (každá trieda a rozhranie vo vlastnom súbore)?

```
public interface I {
   void m();
}

public class M implements I {
   List<I> p;
   public m() {
      ...
      for (I e : p)
         e.m();
   }
}

public class S implements I {
   public m() { ... }
   ...
}
```

(a) Composite
(b) MVC
(c) Visitor
(d) Observer
(e) Strategy

Objektovo-orientované programovanie **B**
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Test – 4. apríl 2017

10 b

1 pay pax pbx pby

2 c

3 b

4 c

5 e

6 a

7 d

8 e

9 a