Objektovo-orientované programovanie **B**

doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Test – 15. apríl 2016

Priezvisko:	tlačeným písmom
Meno:	

1b	
2b	

Test trvá 30 minút.

1	
2	
3	
4	
5	
6	
7	
8	
9	

Odpovede na otázky vpíšte do tabuľky. Hodnotia sa len odpovede v tabuľke. Otázky sú s výberom odpovede, z ktorých 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. (1 b) Okrem iného, program zabezpečuje výpočet podľa určitého vzorca. Používateľ tento výpočet nakoniec spúšťa kliknutím na tlačidlo v používateľskom rozhraní realizovanom v rámci Swing. Samotný výpočet by z hľadiska objekovo-orientovaného návrhu bolo najlepšie realizovať

(a) v metóde `main`()
(b) ako metódu okna, v ktorom sa nachádza tlačidlo
(c) v zodpovedajúcej triede aplikačnej logiky
(d) ako metódu tlačidla
(e) v implementácii prijímača (listener) tlačidla

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

```
interface I {
   void m();
}
abstract class C implements I {
   public void m() {
      System.out.print("c");
   }
}
class D extends C {
   public void m() {
      super.m();
      System.out.print("d");
   }
}
class E extends D {
   public void m() {
      super.m();
      System.out.print("e");
   }
}
class M {
   public static void exe(I... a) {
      for (I e : a)
         e.m();
   }
   public static void main(String[] args) {
      E o1 = new E();
      I o2 = new D();
      C o3 = new D();
      I o4 = (I) new E();

      exe(o1, (I)o2, o3, o4);
   }
}
```

3. (1 b) Agregácia v objektovo-orientovanom programovaní

(a) stanovuje kritéria pre použitie abstraktných tried
(b) znamená spájanie objektov do väčších celkov
(c) umožňuje, aby sa objekt uplatnil namiesto objektu jeho nadtypu
(d) znamená skrytie implementácie objektu
(e) predstavuje kritérium pre použitie dedenia

4. (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
      ;
}
```

Na odstránenie hlavného problému, ktorý bráni tomu, aby tento kód bol v súlade s objektovo-orientovanými princípmi, bolo by potrebné použiť

(a) atribúty
(b) preťaženie metód
(c) modifikátory prístupu
(d) statické metódy
(e) prekonávanie metód

5. (1 b) Pre ktorý návrhový vzor je tento kód charakteristický (každá trieda a rozhranie vo vlastnom súbore)?

```
public interface P {
   void m(N d);
}

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

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

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

public class X implements N {
   public void op(A e) { ... }
   public void op(B e) { ... }
}

public class Y implements N {
   public void op(A e) { ... }
   public void op(B e) { ... }
}
```

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

6. (1 b) V objektovo-orientovanom programe hlavná funkcionalita typicky

(a) je obsiahnutá v metóde `main`()
(b) vzniká v interakcií objektov
(c) je zabezpečená statickými metódami
(d) vzniká dedením
(e) je obsiahnutá v konštruktoroch

7. (1 b) Niť v Jave vzniká

(a) pre každú metódu označenú kľúčový slovom **synchronized**
(b) automaticky z každej metódy
(c) procesom opačným k serializácii
(d) priamym vyvolaním príslušného mechanizmu Java API
(e) automaticky pre každú triedu

8. (1 b) Daná je trieda

```
class C implements Serializable {
   String[] s = new String[9999];
}
```

Dané sú inštancie triedy `C`:

```
C a = new C();
C b = new C();
C c = new C();
```

Serializovať tieto inštancie naraz (jedným zápisom do jedného súboru)

(a) nebude možné, lebo trieda `C` nie je finálna
(b) nebude možné, lebo nie sú prepojené
(c) bude možné
(d) nebude možné, lebo trieda `C` nie je označená ako **public**
(e) nebude možné preto, že trieda `C` obsahuje príliš veľký atribút `s`

9. (1 b) Vyhodenie výnimky v Jave

(a) signalizuje výnimočnú situáciu klientskemu kódu
(b) uvoľňuje pamäť od zbytočných objektov
(c) opravuje vzniknutú chybu
(d) signalizuje výnimočnú situáciu hlavnej metóde v programe
(e) posiela správu vedúcemu programátorovi

Objektovo-orientované programovanie **B**

doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Test – 15. apríl 2016

10 b

1 c

2 cdecdcdcde

3 b

4 e

5 c

6 b

7 d

8 b

9 a