

Objektovo-orientované programovanie

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

Test – náhradný termín – 10. máj 2017

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

2. (1 b) Anonymné triedy sa v Jave typicky používajú

- (a) na ochranu údajov
- (b) na zaradenie aplikačnej funkcionality do rámca
- (c) aby pri kompilácii nevznikalo zbytočne veľa *class* súborov
- (d) na znemožnenie tvorby objektov
- (e) na utajenie názvov tried

3. (1 b) Synchronizované metódy v Jave

- (a) predstavujú kritické regióny programu
- (b) bránia uviaznutiu
- (c) sa rovnomerne striedajú
- (d) sa spúšťajú a končia naraz
- (e) predstavujú nite programu

4. (1 b) Daný je nasledujúci kód v Jave:

```
class MyException extends Exception {}

class A {
    void a(int i) throws MyException {
        if (i > 0) {
            ...
        } else
            throw new MyException();
    }
}

class B {
    void b(int i) {
        new A().a(i);
    }
}
```

Aby bola korektná z pohľadu výnimiek, metóda `b()` triedy `B` musí

- (a) vyhadzovať výnimku typu `MyException` v bloku `try`
- (b) vyhadzovať výnimku typu `MyException` v bloku `finally`
- (c) obsahovať klauzulu `throws MyException` a zároveň aj blok spracovania pre výnimky typu `MyException`
- (d) obsahovať klauzulu `throws MyException` alebo blok spracovania výnimky typu `MyException`
- (e) vyhadzovať výnimku typu `MyException` v bloku `catch`

5. (1 b) Zapuzdrenie v objektovo-orientovanom programovaní

- (a) umožňuje znížiť závislosť klientskeho kódu
- (b) umožňuje spájanie objektov
- (c) predstavuje spôsob tvorenia hierarchie
- (d) umožňuje, aby sa objekt uplatnil namiesto objektu jeho nadtypu
- (e) predstavuje kritérium pre použitie agregácie

6. (1 b) Ak `Objednavka` predstavuje triedu v programe v Jave, výraz `Objednavka.class` predstavuje referenciu na

- (a) súbor, ktorý reprezentuje triedu `Objednavka`
- (b) názov triedy `Objednavka` typu `String`
- (c) inštanciu triedy `Objednavka`
- (d) objekt, ktorý reprezentuje triedu `Objednavka`
- (e) atribút triedy `Objednavka`

7. (1b) 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) Visitor
- (b) Observer
- (c) MVC
- (d) Strategy
- (e) Composite

8. (2b) 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);
    }
}
```

9. (1b) Daná je nasledujúca trieda v Jave:

```
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");
```

```
a.m(b);
b.m(c, d);
c.m(b);
d.m(b, c, d);
```

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

Objektovo-orientované programovanie

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

Test– náhradný termín – 10. máj 2017

10 b

1 d

2 b

3 a

4 d

5 a

6 d

7 e

8 cdecdecde

9 c