

Objektovo-orientované programovanie

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

Test – 17. apríl 2015

B

Priezvisko:	tlačeným písmom
Meno:	

1b	
2b	

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Test trvá 30 minút.

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) Modifikátory prístupu v Jave umožňujú

- (a) redukovať počet atribútov
- (b) regulovať, či iní môžu vidieť kód a zasahovať doň
- (c) redukovať závislosti v kóde
- (d) regulovať, čo sa ukladá pri serializácii
- (e) redukovať počet metód

2. (1 b) Okrem iného, program zabezpečuje zistenie počtu výskytov jedného reťazca znakov v inom reťazci znakov, čo sa spúšťa kliknutím na tlačidlo. Samotné zistenie počtu výskytov by bolo najlepšie realizovať

- (a) ako metódu okna, v ktorom sa nachádza tlačidlo
- (b) ako metódu v triede mimo používateľského rozhrania
- (c) ako metódu tlačidla
- (d) v implementácii prijímača (listener) tlačidla
- (e) v metóde `main()`

3. (1 b) V rámci Swing sa kliknutie na tlačidlo v okne sleduje

- (a) zavolaním statickej metódy `EventQueue.onClick()` a následným zistením, či sa kliknutie vzťahuje na dané tlačidlo
- (b) automaticky po pridaní tlačidla do okna metódou `add()`
- (c) prijímačom udalosti kliknutia registrovaným pre dané tlačidlo
- (d) volaním metódy `actionPerformed()` príslušného tlačidla v slučke
- (e) metódou `onClick()` tlačidla implementovanou pri odvození od všeobecného tlačidla `JButton`

4. (2 b) Daný je nasledujúci kód v Jave (každá trieda vo vlastnom súbore):

```
public class A {
    public void m() { System.out.print("a."); }
}
```

```
public class B extends A {
    public void m() { System.out.print("b."); }
}
```

```
public class X {
    public void op(A o) {
        System.out.print("x");
        o.m();
    }
    public static void main(String[] args) {
        A o1 = new A();
        B o2 = new B();
        A o3 = new B();

        X x = new X();
        Y y = new Y();
        X z = new Y();

        x.op(o1);
        x.op(o2);
        x.op(o3);

        y.op(o1);
        y.op(o2);
        y.op(o3);

        z.op(o1);
        z.op(o2);
        z.op(o3);
    }
}
```

```
public class Y extends X {
    public void op(A o) {
        System.out.print("y");
        o.m();
    }
}
```

Čo sa vypíše vykonaním triedy X?

- (a) xa.xb.xa.ya.yb.ya.ya.yb.ya.
- (b) xa.xa.xa.ya.ya.ya.ya.ya.
- (c) xa.xb.xb.ya.yb.yb.ya.yb.yb.
- (d) xa.xa.xa.ya.ya.ya.xa.xa.xa.
- (e) xa.xb.xa.ya.yb.ya.xa.xb.xa.

5. (1 b) Zavedenie (import) viacerých balíkov v Jave môže spôsobiť

- (a) riešiteľnú kolíziu názvov zavedených prvkov
- (b) zvýšenie veľkosti výsledného JAR súboru
- (c) spomalenie programu
- (d) zvýšenie veľkosti class súborov tried, ktoré používajú prvky príslušného balíka
- (e) neriešiteľnú kolíziu názvov zavedených prvkov

6. (2b) Pre ktorý návrhový vzor je tento kód charakteristický (každá trieda a rozhranie vo vlastnom súbore)?

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

public class X implements I {
    List<I> list;
    public doSomething() {
        ...
        for (I e : list)
            e.doSomething();
    }
    ...
}

public class Y implements I {
    public doSomething() { ... }
    ...
}
```

- (a) Strategy
- (b) Visitor
- (c) Observer
- (d) Composite
- (e) žiaden

7. (1b) Na dodržanie Liskovej princípu substitúcie v prekonávajúcich metódach treba dbať na

- (a) súdržnosť a zviazanosť
- (b) typ návratovej hodnoty
- (c) synchronizáciu a implicitné nite
- (d) počet riadkov
- (e) implicitné predpoklady a dôsledky

8. (1b) Synchronizované metódy v Jave

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

9. (1b) Pri serializácii skupiny objektov do jedného súboru je kľúčové

- (a) aby sa žiaden objekt v skupine neodkazoval na iné jej objekty
- (b) aby atribúty objektov v skupine mali viditeľnosť `public`
- (c) aby objekty v skupine neboli príliš veľké
- (d) aby v skupine bol objekt, z ktorého sa dá dostať k jej ostatným objektom
- (e) aby sa medzi objektmi v skupine nevyskytovali cirkulárne prepojenia

10. (2b) Ak sa v programe vyskytuje nasledujúci kód:

```
void aktivujPredmet(Object o) {
    if (MagickyPrsten.class.isInstance(o))
        ((MagickyPrsten)o).zasviet();
    else if (CarovnyPrutik.class.isInstance(o))
        ((CarovnyPrutik)o).vycaruj();
    else if ...
}
```

ten program porušuje

- (a) Liskovej princípu substitúcie
- (b) pravidlá dedenia
- (c) princípu zapuzdrenia
- (d) pravidlá polymorfizmu
- (e) princípu otvorenosti a uzavretosti

11. (1b) Anonymné triedy sa v Jave typicky používajú

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

12. (1b) Prúd údajov (stream) v Java API sa otvára

- (a) jeho konštruktorom
- (b) príkazom `IOStream.open()`
- (c) jeho prvým použitím
- (d) jeho metódou `open()`
- (e) príkazom `System.open()`

15 b

1 c

2 b

3 c

4 c

5 a

6 d

7 e

8 a

9 d

10 e

11 d

12 a