

Objektovo-orientované programovanie

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

Semestrálny test – 12. apríl 2013

A

Priezvisko:

1b	
2b	

Meno:

Test trvá 35 minút.

	a	b	c	d	e
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

V otázkach je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do tabuľky. Hodnotia sa len odpovede v tabuľke. V prípade opravy jasne vyznačte odpoveď, ktorá platí. Každá správna odpoveď má hodnotu vyznačenú v otázke. Nesprávna odpoveď, vyznačenie viac odpovedí alebo nejednoznačné vyznačenie má hodnotu 0 bodov. Postup riešenia sa nehodnotí. Len celistvý list bude akceptovaný.

1. (1 b) Vyhodená výnimka v Jave je

- (a) trieda
- (b) rozhranie
- (c) objekt
- (d) metóda
- (e) atribút

2. (1 b) Prúd údajov (stream) v Java API sa otvára

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

3. (1 b) Použitie príkazu `import` na import balíkov v Jave

- (a) zvýši veľkosť JAR súboru programu
- (b) neovplyvní veľkosť programu
- (c) zvýši veľkosť class súborov tried, ktoré sa nachádzajú v tom istom súbore ako príkaz `import`
- (d) zníži veľkosť class súborov tried, ktoré používajú prvky príslušného balíka
- (e) zvýši veľkosť class súborov tried, ktoré používajú prvky príslušného balíka

4. (1 b) Synchronizované metódy v Jave

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

5. (2 b) Na dodržanie Liskovej princípu substitúcie v prekonávajúcich metódach treba dbať na

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

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

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

7. (1 b) Atribút triedy, ktorému predchádza kľúčové slovo `private`

- (a) sa nezapíše do súboru pri serializácii objektu
- (b) je dostupný len v danej hierarchii tried
- (c) je dostupný len v danej triede
- (d) je dostupný len v rámci jednej nite
- (e) je chránený pred zápisom

8. (2 b) Daná je trieda

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

Dané sú inštancie triedy `Exp`:

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

Serializovať tieto inštancie naraz (jedným zápisom do jedného súboru)

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

9. (2b) Daný je nasledujúci kód v Jave:

```
class X {
    public void m() { System.out.print("x"); }
}
class Y extends X {
    public void m() { System.out.print("y"); }
}
class Z extends Y {
    public void m() { System.out.print("z"); }
}
class M {
    public static void op(X o) { o.m(); }
}
```

Čo sa vypíše po vykonaní týchto príkazov:

```
X o1 = new X();
X o2 = new Z();
Y o3 = new Y();
```

```
M.op(o1);
M.op((X)o1);
M.op(o2);
M.op((X)o2);
M.op(o3);
M.op((X)o3);
```

- (a) xyzxyz
- (b) xxxzzy
- (c) xxxxyy
- (d) xxxzyx
- (e) xxxxyx

10. (1b) Agregácia v objektovo-orientovanom programovaní

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

11. (2b) 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);
    }
}
```

Metóda b() triedy B

- (a) musí v časti **try** bloku spracovania výnimky vyhadzovať výnimku typu **MyException**
- (b) musí v časti **finally** bloku spracovania výnimky vyhadzovať výnimku typu **MyException**
- (c) je korektná z pohľadu výnimiek
- (d) musí obsahovať klauzulu **throws MyException** alebo blok spracovania výnimky typu **MyException**
- (e) musí obsahovať klauzulu **throws MyException** a zároveň aj blok spracovania výnimky typu **MyException**
- (f) musí v časti **catch** bloku spracovania výnimky vyhadzovať výnimku typu **MyException**

15 b

1 c

2 a

3 b

4 d

5 d

6 a

7 c

8 e

9 b

10 e

11 d