

Objektovo-orientované programovanie

Ing. Valentino Vranič, PhD., ÚISI FIIT STU

Semestrálny test — 3. apríl 2009

B

Priezvisko:	
Meno:	

1b	
2b	

	a	b	c	d	e
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

Test trvá 50 minút.

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. List odovzdajte nepoškodený.

1. (1 b) Porušenie Liskovej princípu substitúcie spôsobí

- (a) zvýšenie zviazania tried
- (b) inverziu generalizácie a špecializácie
- (c) nekonečnú rekurziu
- (d) neočakávané správanie klientskeho kódu
- (e) chybu pri preklade

2. (2 b) Daný je nasledujúci kód:

```
List<R> z = new ArrayList<R>();  
z.add(new X());  
z.add(new Y());
```

Aby sa tento kód mohol preložiť a vykonať

- (a) typ X musí byť odvodený od typu Y alebo naopak
- (b) typy X a Y musia byť triedy priamo alebo nepriamo odvodené od typu R
- (c) typy X a Y musia byť triedy odvodené od typu R, ale X nesmie byť odvodené od Y alebo naopak
- (d) R musí byť abstraktná trieda, a typy X a Y triedy od nej odvodené
- (e) R musí byť rozhranie, a typy X a Y triedy, ktoré ho implementujú

3. (1 b) Je dobré z hľadiska objektovo-orientovaného prístupu rozsiahle používanie statických metód?

- (a) Áno, lebo umožňujú polymorfizmus.
- (b) Nie, lebo sa nededia.
- (c) Nie, lebo sa nedajú prekonať.
- (d) Áno, lebo sa vykonávajú rýchlejšie.
- (e) Nie, lebo nepodporujú zapuzdrenie.

4. (1 b) Uplatnenie objektu namiesto objektu typu, od ktorého je odvodený, je možné vďaka

- (a) polymorfizmu
- (b) súdržnosti
- (c) perzistencii
- (d) zapuzdreniu
- (e) agregácii

5. (1 b) Zmeniť stav objektu znamená

- (a) zmeniť jeho inicializáciu
- (b) zmeniť jeho metódy
- (c) zmeniť zoznam jeho atribútov
- (d) zmeniť hodnoty jeho atribútov
- (e) zmeniť jeho konštruktor

6. (2 b) Daný je nasledujúci kód v Jave:

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

Čo sa vypíše po vykonaní týchto príkazov:

```
A b = new B();  
A c = new C();
```

```
b.m();  
((A)b).m();  
c.m();  
((B)c).m();
```

- (a) acc
- (b) aaaa
- (c) aacc
- (d) cc
- (e) aaca

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

- (a) je dostupný vo vlastnej a odvodených triedach
- (b) je dostupný len v rámci nite, v ktorej bol vytvorený objekt
- (c) sa nezapíše do súboru pri serializácii objektu
- (d) je chránený pred zápisom
- (e) je dostupný len vo vlastnej triede

8. (1 b) Dané je rozhranie:

```
interface Abc {  
    void m();  
}
```

Z pohľadu prekladu v nasledujúcom kóde

```
Abc o;  
o = new Abc();  
o.m();
```

- (a) len prvý riadok je zlý
- (b) len druhý riadok je zlý
- (c) len tretí riadok je zlý
- (d) druhý a tretí riadok sú zlé
- (e) zlé sú všetky tri riadky

9. (1 b) Ak potrebujeme pracovať s V/V systémom Javy, príkaz

```
import java.io.*;
```

- (a) nie je nevyhnutný, ale potrebné triedy V/V systému musíme skopírovať do vlastného programu
- (b) nie je nevyhnutný, ale `java.io.*` musíme zadať pri kompilácii
- (c) je nevyhnutný
- (d) je zbytočný
- (e) nie je nevyhnutný

10. (1 b) Prúd údajov (stream) v Java API sa otvára

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

11. (1 b) Iterátory v Java API uľahčujú

- (a) rušenie prvkov v zoskupeniach
- (b) volania abstraktných metód
- (c) opakovanie vykonávania ľubovoľného kódu
- (d) pridávanie prvkov do zoskupení
- (e) prechádzanie zoskupeniami

12. (2 b) Metóda `f()` triedy `A` vyhadzuje výnimku `MyException`. Daná je trieda `B`:

```
class B {  
    void m() { new A().f(); }  
}
```

Metóda `m()` triedy `B`

- (a) musí vyhadzovať výnimku typu `MyException`
- (b) musí deklarovať, že vyhadzuje výnimku typu `MyException`
- (c) musí deklarovať alebo ošetrovať výnimku typu `MyException`
- (d) je korektná
- (e) musí ošetrovať výnimku typu `MyException`

15 b

1 d

2 b

3 c

4 a

5 d

6 c

7 e

8 b

9 e

10 a

11 e

12 c