Objektovo-orientované programovanie **B**

Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Semestrálny test — 27. marec 2007

Priezvisko:		1b	
Meno:		2b	

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

Test trvá 60 minút.
V otázkach je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do veľkej tabuľky (malú tabuľku nevypĺňajte). Hodnotia sa len odpovede vyznačené v tabuľke.
V prípade opravy jasne vyznačte odpoveď ktorú vyberáte. Každá správna odpoveď má hodnotu vyznačenú v otázke. Nesprávna odpoveď, vyznačenie viac odpovedí alebo nejednoznačné vyznačenie má hodnotu 0 bodov. Postup riešenia sa nehodnotí.

1. (2 b) Daný je nasledujúci program v Jave:

```
public class C {
   public void m() {
      System.out.print("abc");
   }
   public static void main(String[] args) {
      new C(){
         public void m(){
            System.out.print("xyz");
         }
      }.m();
   }
}
```

Pri spustení triedy C

(a) vypíše sa abc
(b) vypíše sa xyzabc
(c) vypíše sa abcxyz
(d) vypíše sa xyz
(e) vznikne chyba pri vykonávaní

2. (1 b) Princíp otvorenosti a uzavretosti hovorí, že

(a) tok údajov pred čítaním treba otvoriť, ale pred skončením programu uzavrieť
(b) softvérové entity majú byt otvorené pre rozšírenie, ale uzavreté pre zmeny
(c) správanie objektu môže byť otvorené, ale jeho implementácia má byť uzavretá
(d) tok údajov pred čítaním treba otvoriť, a tým sa automaticky zabezpečí aj jeho uzavretie po skončení programu
(e) softvérové entity majú byt voľne zviazané, ale vysoko súdržne

3. (1 b) Tok údajov (stream) v Java API sa obaľuje, aby

(a) prístup k toku údajom mohol byť realizovaný operáciami vyššej úrovne
(b) bolo vôbec možné pracovať s tokom údajov
(c) nedošlo k strate údajov v dôsledku vysokej rýchlosti ich prúdenia tokom údajov
(d) bolo ľahšie zachytiť výnimku IOException
(e) bolo možné tok údajov presmerovať

4. (1 b) Prvky tried, pre ktoré nie je uvedený prístup jedným z modifikátorov, sú implicitne

(a) prístupné len v triedach toho istého balíka
(b) private
(c) prístupné len v odvodených triedach a v triedach toho istého balíka
(d) protected
(e) public

5. (2 b) Daný je nasledujúci program v Jave:

```
class X {
   interface I {
      void m();
   }

   public static void main(String[] args) {

      abstract class A implements I { }

      class B extends A {
         public void m() { }
      }

      I[] a = new A[] {new B(), new B ()};
   }
}
```

Tento program sa

(a) nepreloží, lebo sa referencii a priraďuje objekt, ktorý jej nezodpovedá
(b) preloží a vykoná korektne
(c) preloží, ale vznikne chyba pri vykonávaní posledného riadku
(d) nepreloží, lebo nie je možné definovať triedy v metódach
(e) nepreloží, lebo trieda A neimplementuje metódu m()

6. (1 b) V programe v Jave príkaz

```
package abc.xyz;
```

(a) sprístupňuje všetky typy definované v balíku abc.xyz
(b) sprístupňuje všetky typy definované v balíku abc.xyz okrem typov v podbalíkoch
(c) deklaruje balík abc.xyz a môže sa vyskytovať len v jednom súbore
(d) sprístupňuje triedu xyz z balíka abc
(e) deklaruje balík abc.xyz a môže sa vyskytovať v ľubovoľnom počte súborov

7. (1 b) Daný je nasledujúci kód v Jave:

```
List<Element> z = new ArrayList<Element>();

for (Element e : z)
    e.print();
```

V uvedenej slučke **for** sa *implicitne* využíva objekt typu:

(a) List
(b) Element
(c) Separator
(d) Iterator
(e) Alternator

8. (1 b) Daný je kód v Jave na obr. 1. Vytvoríme nasledujúci objekt:

```
M o = new N();
```

Pri vykonaní týchto príkazov:

```
o.f();
(N)o.f();
((N)o).f();
```

(a) vznikne výnimka
(b) vypíše sa mnm
(c) vypíše sa nnn
(d) vypíše sa mnn
(e) vypíše sa mmn

```
class M {
    public M() {
        System.out.print("M");
    }
    public void f() {
        System.out.print("m");
    }
}

class N extends M {
    public N() { System.out.print("N"); }
    public void f() {
        System.out.print("n");
    }
}
```

Obrázok 1: Kód pre otázky 8 a 10.

9. (1 b) Abstraktná trieda v Jave

(a) môže mať len abstraktné metódy
(b) nemôže mať polia
(c) nemôže dediť
(d) nemôže mať prekonávajúce metódy
(e) môže mať statické metódy

10. (2 b) Daný je kód v Jave na obr. 1. Pri vykonaní týchto príkazov:

```
M o1, o2;
o1 = new M();
o2 = new N();
```

(a) vypíše sa MNN
(b) vypíše sa MMN
(c) vypíše sa MN
(d) vznikne výnimka
(e) vypíše sa MNMN

11. (1 b) Nech m1() je metóda triedy, ktorá vracia Integer. Reťazec a je v jednej z inštančných (nestatických) metód tej istej triedy definovaný takto:

```
String a[] = new String[m1()];
```

Táto definícia je v Jave

(a) korektná
(b) nekorektná
(c) korektná jedine ak je metóda m1() statická
(d) korektná jedine ak je metóda m1() synchronizovaná
(e) korektná jedine ak je metóda m1() finálna a statická

12. (1 b) Dá sa v Jave urobiť *inštancia* rozhrania?

(a) áno
(b) áno, ale len ak neobsahuje metódy
(c) nie
(d) áno, ale len ak obsahuje výlučne statické prvky
(e) áno, ale nebudú sa dať zavolať jeho metódy

Objektovo-orientované programovanie **B**
Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Semestrálny test — 27. marec 2007

15 b

1 d

2 b

3 a

4 a

5 b

6 e

7 d

8 a / c (chyba v otázke)

9 e

10 b

11 a

12 c