

## Objektovo-orientované programovanie

Ing. Valentino Vranič, PhD., ÚISI FIIT STU

Semestrálny test — 27. marec 2007

A

Priezvisko:		1b	
Meno:		2b	

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.

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

1. (1 b) V programe v Jave príkaz

```
package abc.xyz;
```

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

2. (1 b) Tok údajov (stream) v Java API sa obaľuje, aby

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

3. (1 b) Nech `m1()` je metóda triedy, ktorá vracia `Integer`. Referenciu `a` je v jednej z inštančných (nestatických) metód tej istej triedy definovaný takto:

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

Táto definícia je v Jave

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

4. (1 b) Dá sa v Jave urobiť inštancia rozhrania?

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

5. (1 b) Prvky tried, pre ktoré nie je uvedený prístup jedným z modifikátorov, sú implicitne

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

6. (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 nie je možné definovať triedy v metódach
- (b) preloží, ale vznikne chyba pri vykonávaní posledného riadku
- (c) nepreloží, lebo trieda A neimplementuje metódu `m()`
- (d) nepreloží, lebo sa referenciou a priraduje objekt, ktorý jej nezodpovedá
- (e) preloží a vykoná korektné

7. (1 b) Princíp otvorenosti a uzavretosti hovorí, že

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

8. (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) Separator
- (b) Alternator
- (c) Iterator
- (d) Element
- (e) List

9. (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 MNMN
- (b) vypíše sa MN
- (c) vznikne výnimka
- (d) vypíše sa MMN
- (e) vypíše sa MNN

```
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 9 a 11.

10. (1 b) Abstraktná trieda v Jave

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

11. (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) vypíše sa mnn
- (b) vznikne výnimka
- (c) vypíše sa mnn
- (d) vypíše sa mnn
- (e) vypíše sa mnn

12. (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 xyz
- (c) vznikne chyba pri vykonávaní
- (d) vypíše sa xyzabc
- (e) vypíše sa abcxyz

15 b

1 c

2 c

3 d

4 a

5 e

6 e

7 d

8 c

9 d

10 d

11 a / b (chyba v otázke)

12 b