Objektovo-orientované programovanie **B**

Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Semestrálny test — 3. apríl 2006

Priezvisko:		**Body**
Meno:		

	a	b	c	d	e
1					
2					
3					
4					
5					
7					
8					
9					
10					
11					
12					
13					
14					
15					

V otázkach je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do tabuľky uvedenej nižšie. Hodnotia sa len odpovede vyznačené v tabuľke. V prípade opravy jasne vyznačte ktorú odpoveď vyberáte.
Každá správna odpoveď má hodnotu 1 bod. Nesprávna odpoveď, vyznačenie viac odpovedí alebo nejednoznačné vyznačenie má hodnotu 0 bodov. Postup riešenia sa nehodnotí.

1. Objekt v objektovo-orientovanom programovaní predstavuje

(a) triedu
(b) inštanciu triedy
(c) inštanciu triedy alebo rozhrania
(d) typ
(e) modul

2. Príkaz

import java.util.∗;

(a) fyzicky pripojí typy balíka java.util k programu bez typov v podbalíkoch
(b) fyzicky pripojí len skutočne použité typy balíka java.util k programu bez typov v podbalíkoch
(c) fyzicky pripojí všetky typy balíka java.util k programu vrátane typov v podbalíkoch
(d) sprístupní priestor názvov všetkých typov balíka java.util vrátane typov v podbalíkoch
(e) sprístupní priestor názvov všetkých typov balíka java.util, ale bez typov v podbalíkoch

3. Prístup **protected** je vhodné použiť pri takých prvkoch triedy ku ktorým chceme pristupovať len

(a) v triedach toho istého balíka
(b) v danej triede
(c) v odvodených triedach
(d) v odvodených triedach a v triedach toho istého balíka
(e) v odvodených triedach toho istého balíka

4. Tok údajov (stream) v Java API sa otvára

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

5. Nech o je objekt triedy ktorá poskytuje metódu **int** m(). Reťazec r je definovaný takto:

Integer r[] = **new** r[o.m()];

Táto definícia je

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

6. Dá sa urobiť inštancia abstraktnej triedy?

(a) áno, ale len ak trieda neobsahuje abstraktné metódy
(b) áno, ako hociktorej inej triedy
(c) nie
(d) áno, ale bude abstraktná
(e) áno, ale nebudú sa dať zavolať abstraktné metódy

7. Daný je kód v Jave na obr. 1. Dá sa z metódy s() triedy X zavolať rovnomenná metóda triedy C?

(a) áno, príkazom X.s();
(b) áno, príkazom s();
(c) áno, príkazom super.s();
(d) nie
(e) áno, príkazom this.s();

8. Daný je kód v Jave na obr. 1. Čo sa vypíše po vykonaní týchto príkazov:

```
C o = new X();
o.m();
o.s();
((X)o).s();
```

(a) Xm Xs Xs
(b) Cm Cs Xs
(c) Cs Xm Cs
(d) Xm Cs Xs
(e) Cm Xs Xs

```
class C {
    void m() { System.out.print("Cm "); }
    static void s() { System.out.print("Cs "); }
}

class X extends C {
    void m() { System.out.print("Xm "); }
    static void s() { System.out.print("Xs "); }
}
```

Obr. 1: Kód pre otázky 8, 7 a 11.

9. Daný je nasledujúci kód:

```
for (Object o : l) {
    if (o.class == Frog.class)
        ((Frog)o).jump();
    else if (o.class == Bird.class)
        ((Bird)o).fly();
    else
        ;
}
```

Tento kód porušuje

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

10. Konštruktor v Jave

(a) nemusí byť explicitne poskytnutý
(b) môže vraciať ľubovoľnú hodnotu
(c) nemôže byť preťažený
(d) nemôže mať argumenty
(e) musí mať vždy len jeden argument

11. K triedam z obr. 1 je daný nasledujúci kód:

```
List<C> list = new ArrayList<C>();
list.add(new X());
```

Tento kód sa

(a) nepreloží, lebo typ referencie list nezodpovedá typu priradeného objektu
(b) nepreloží, lebo do zoznamu list sa dajú vkladať len objekty typu C
(c) preloží a vykoná korektne
(d) preloží, ale padne počas vykonávania s výnimkou ClassCastException
(e) nepreloží, lebo trieda ArrayList nie je generická

12. Daný je nasledujúci kód:

```
class MyException extends Exception {}

class A {
    void m() throws MyException {
        . . .
    }
}

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

Metóda m() triedy B

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

13. Kľúčové slovo **synchronized** slúži na

(a) uzamknutie objektu pre vyhradený prístup k jeho poliam
(b) zmenu priority vykonávania nite
(c) zmenu poradia vykonávania nití
(d) spojenie dvoch niti
(e) určenie metód ktoré sa môžu vykonávať súčasne

14. Daný je nasledujúci kód:

```
1  abstract class A { }
2  interface I { }
3  A[] a = new A[5];
4  I[] i = new I[5];
```

Tento kód sa

(a) preloží a vykoná korektne
(b) nepreloží, lebo prekladač hlási chybu na riadku 3
(c) nepreloží, lebo prekladač hlási chybu na riadku 4
(d) preloží, ale vznikne chyba pri vykonávaní riadku 3
(e) preloží, ale vznikne chyba pri vykonávaní riadku 4

15. Daný je nasledujúci program:

```
**1** B {
    **2**
}

public class A {
    **3** f() {
        return new **3**() {
            public void m() {
                System.out.println("M");
                }
        };
    }
    public static void main(String[] args) {
        new A().f().m();
    }
}
```

Ktoré fragmenty kódu treba v tomto programe doplniť, aby sa pri jeho vykonaní vypísalo M?

(a) `**1**`: **class** `**2**`: **void** f(); `**3**`: B
(b) `**1**`: **class** `**2**`: **void** m(); `**3**`: A
(c) `**1**`: **class** `**2**`: **void** m(); `**3**`: B
(d) `**1**`: **interface** `**2**`: **void** m(); `**3**`: B
(e) `**1**`: **interface** `**2**`: **void** f(); `**3**`: B

Objektovo-orientované programovanie **B**
Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Semestrálny test — 3. apríl 2006

1 b

2 e

3 d

4 e

5 c

6 c

7 a

8 d

9 d

10 a

11 c

12 b

13 a

14 a

15 d

Body: 15