Objektovo-orientované programovanie **B**

Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Semestrálny test — 4. apríl 2008

Priezvisko:		1b	
Meno:		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 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. (1 b) Dá sa urobiť inštancia abstraktnej triedy?

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

2. (1 b) V triede, ktorá implementuje rozhranie, je možné zadefinovať

(a) len metódy nedeklarované v rozhraní
(b) len ďalšie polia
(c) ľubovoľné ďalšie polia a metódy
(d) len ďalšie metódy
(e) len metódy deklarované v rozhraní

3. (1 b) Zapuzdrenie v objektovo-orientovanom programovaní

(a) umožňuje, aby sa objekt uplatnil namiesto objektu jeho nadtypu
(b) umožňuje znížiť závislosť klientskeho kódu
(c) umožňuje spájanie objektov
(d) predstavuje spôsob tvorenia hierarchie
(e) predstavuje kritérium pre použitie agregácie

4. (1 b) Daný je nasledujúci kód:

```
if (o.class == "A")
    ((A)o).ma();
else if (o.class == "B")
    ((B)o).mb();
else
    ...
}
```

Tento kód porušuje

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

5. (2 b) Daný je nasledujúci kód:

```
class MyException extends Exception {}

class A {
    void a() throws MyException {
        if (...) {
            ...
        }
        else
            throw new MyException();
    }
}

class B {
    void b() throws MyException {
        new A().a();
    }
}
```

Metóda b() triedy B

(a) musí vyhadzovať výnimku typu MyException
(b) je korektná
(c) musí ošetrovať výnimku typu MyException
(d) musí zachytávať výnimku typu MyException
(e) nesmie obsahovať klauzulu **throws**

6. (1 b) Objekt v objektovo-orientovanom programovaní predstavuje

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

7. (1 b) Prístup **protected** je vhodné použiť pri takých prvkoch triedy, ku ktorým chceme pristupovať len

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

8. (1 b) Príkaz

```
import java.util.*;
```

(a) sprístupní priestor názvov všetkých typov balíka java.util vrátane 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) fyzicky pripojí typy balíka java.util k programu bez typov v podbalíkoch
(e) sprístupní priestor názvov všetkých typov balíka java.util, ale bez typov v podbalíkoch

9. (2 b) Čím treba nahradiť označené časti kódu z obr. 1, aby metóda add() pripájala ďalší prvok so zadaným údajom data k aktuálnemu prvku pre rôzne typy údajov, ako ukazuje metóda main().

(a) `**1**`: `**2**`: T d `**3**`: El(d)
(b) `**1**`: <data> `**2**`: data d `**3**`: El<data>(d)
(c) `**1**`: <T> `**2**`: T d `**3**`: El<T>(d)
(d) `**1**`: <T> `**2**`: <T> d `**3**`: El(<T> d)
(e) `**1**`: `**2**`: d `**3**`: El<d>(d)

```
public class El<T> {
    private T data;
    private El**1** next;

    public El(**2**) {
        data = d;
    }
    public void add(**2**) {
        next = new **3**;
    }

    public static void main(String[] args) {
        El<Double> e1 = new El<Double>(8.4);
        e1.add(10.2);

        El<Integer> e2 = new El<Integer>(10);
        e1.add(12);
    }
}
```

Obrázok 1: Kód pre otázku 9.

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

class B extends A {
    public void m() { System.out.print("b"); }
}

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

Obrázok 2: Kód pre otázky 10 a 11.

10. (1 b) Daný je kód v Jave na obr. 2. Dá sa z metódy m() triedy C zavolať rovnomenná metóda triedy A (bez vytvárania ďalšieho objektu)?

(a) nie
(b) áno, príkazom super.m();
(c) áno, príkazom A.m();
(d) áno, príkazom super.super.m();
(e) áno, príkazom super.this.m();

11. (2 b) Daný je kód v Jave na obr. 2. Čo sa vypíše po vykonaní týchto príkazov:

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

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

(a) aaa
(b) bcb
(c) bcc
(d) aab
(e) aac

12. (1 b) V triede, ktorá dedí od rozhrania je možné zadefinovať

(a) len to čo predpisuje rozhranie
(b) aj iné metódy než predpisuje rozhranie
(c) len konkrétne metódy
(d) len ďalšie polia
(e) len metódy ktoré predpisuje rozhranie

Objektovo-orientované programovanie **B**
Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Semestrálny test — 4. apríl 2008

15 b

1 e

2 c

3 b

4 d

5 b

6 a

7 d

8 e

9 c

10 a

11 c

12 b