

Objektovo-orientované programovanie

doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Náhradný semestrálny test – 23. máj 2013

Priezvisko:

1b	
2b	

Meno:

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

Test trvá 35 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. Len celistvý list bude akceptovaný.

1. (1 b) V objektovo-orientovanom programe hlavná funkcionálna typicky

- (a) je obsiahnutá v metóde `main()`
- (b) vzniká v interakcii objektov
- (c) je obsiahnutá v konštruktoroch
- (d) vzniká dedením
- (e) je zabezpečená statickými metódami

2. (1 b) Rozhranie v Jave definuje

- (a) správanie bez implementácie
- (b) implementáciu metód, ktorá sa dá prekonať
- (c) vzťahy dedenia medzi inými rozhraniami
- (d) implementáciu metód, ktorá sa nedá prekonať
- (e) vzťahy dedenia medzi inými triedami

3. (1 b) Daný je kód v Jave:

```
List<Axis> a = new ArrayList<>();
```

Axis predstavuje

- (a) názov zoznamu
- (b) reťazec znakov
- (c) referenciu na údaje uchovávané v zozname
- (d) neznámu premennú
- (e) typ údajov uchovávaných v zozname

4. (1 b) Pomocou kľúčového slova **super** sa v Jave dá zavolať

- (a) prekonávajúca metóda z prekonanej na ľubovoľnej úrovni dedenia
- (b) ľubovoľná metóda nadtypu
- (c) prekonaná metóda z prekonávajúcej na ľubovoľnej úrovni dedenia
- (d) bezprostredne prekonaná metóda z prekonávajúcej
- (e) bezprostredne prekonávajúca metóda z prekonanej

5. (1 b) Daný je nasledujúci kód v Jave:

```
public class A {
    public int f(int i) {
        return 2 * i;
    }
    public void m() {
        int [] a = new int [f(5)];
    }
}
```

Vytvorenie poľa a

- (a) by bolo korektné len ak by metódy `f()` a `m()` boli statické
- (b) by bolo korektné len ak by metóda `f()` bola statická
- (c) nie je korektné, lebo veľkosť musí byť zadaná konštantou
- (d) nie je korektné, lebo veľkosť nemôže byť zadaná metódou
- (e) je korektné

6. (1 b) Je dobré z hľadiska objektovo-orientovaného prístupu rozsiahle používanie statických metód?

- (a) Nie, lebo neumožňujú polymorfizmus.
- (b) Nie, lebo sa nededia.
- (c) Áno, lebo sa vykonajú rýchlejšie.
- (d) Nie, lebo nepodporujú zapuzdrenie.
- (e) Áno, lebo sa dajú prekonať.

7. (2 b) K triedam:

```
abstract class P {
    int f = 0;
    void m() {
        System.out.println(f);
    }
}
class Q extends P {
    void m(int f) {
        this.f = f;
        m();
    }
}
```

je daný nasledujúci kód v Jave:

```
(new Q()).m();
```

Tento kód

- (a) sa vykoná úspešne a vypíše 0
- (b) nevypíše nič a vyhodí výnimku
- (c) sa nebude dať preložiť
- (d) sa vykoná úspešne a nevypíše nič
- (e) vypíše 0 a vyhodí výnimku

8. (1 b) Iterátory v Java API uľahčujú

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

9. (2b) Daný je nasledujúci kód v Jave:

```
while (getObject(o)) {
    if (o instanceof A)
        ((A)o).opa();
    else if (o instanceof B)
        ((B)o).opb();
    else
        ;
}
```

Na odstránenie hlavného problému, ktorý bráni tomu, aby tento kód bol v súlade s objektovo-orientovanými princípmi, bolo by potrebné použiť

- (a) modifikátory prístupu
- (b) prekonávanie metód
- (c) atribúty
- (d) preťaženie metód
- (e) statické metódy

10. (2b) Daný je nasledujúci kód v Jave:

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

```
class B extends A {
    void m() {
        System.out.print("b ");
    }
    static void s() {
        System.out.print("sb ");
    }
}
```

Čo sa vypíše po vykonaní týchto príkazov:

```
A o = new B();
o.m();
o.s();
((B)o).s();
```

- (a) a sa sb
- (b) b sb sb
- (c) a sb sb
- (d) sa b sa
- (e) b sa sb

11. (2b) Daný je nasledujúci kód v Jave:

```
class MyException extends Exception {}

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

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

Metóda m() triedy B

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

Objektovo-orientované programovanie
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Náhradný semestrálny test – 23. máj 2013

15 b

1 b

2 a

3 e

4 d

5 e

6 a

7 a

8 c

9 b

10 e

11 c