

## Objektovo-orientované programovanie

Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Semestrálny test — 15. apríl 2009



Priezvisko:	
Meno:	

1b	
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 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. List odovzdajte nepoškodený.

1. (1 b) Kľúčové slovo **synchronized** slúži na

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

2. (1 b) Správanie objektu je určené

- (a) operáciami, ktoré poskytuje
- (b) hodnotami jeho atribútov
- (c) spôsobom jeho vytvorenia
- (d) jeho atribútmi
- (e) jeho inicializáciou

3. (2 b) Daný je kód v Jave na obr. 1. 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) b sa sb
- (e) sa b sa

4. (1 b) Daný je nasledujúci kód:

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

Tento kód porušuje

```
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 "); }
}
```

Obrázok 1: Kód pre otázky 3, 5 a 6.

- (a) princíp polymorfizmu
- (b) princíp otvorenosti a uzavretosti kódu
- (c) Liskovej princíp substitúcie
- (d) princíp uzavretosti blokov
- (e) princíp generalizácie a špecializácie

5. (1 b) Daný je kód v Jave na obr. 1. Dá sa z metódy **s()** triedy **B** zavolať rovnomená metóda triedy **A**?

- (a) áno, príkazom **super.s()**
- (b) nie
- (c) áno, príkazom **s()**
- (d) áno, príkazom **((A)this).s()**
- (e) áno, príkazom **A.s()**

6. (2 b) K triedam z obr. 1 je daný nasledujúci kód:

```
List<A> list = new ArrayList<A>();
list.add(new B());
```

Tento kód sa

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

7. (1 b) Kód

```
BufferedReader stdin =
    new BufferedReader(
        new InputStreamReader(System.in));
```

zabezpečuje

- (a) vytvorenie štandardného vstupu
- (b) otvorenie štandardného vstupu
- (c) pohodlnejšiu prácu so štandardným vstupom
- (d) vytvorenie a otvorenie štandardného vstupu
- (e) čítanie zo štandardného vstupu

8. (1 b) Abstraktná trieda v Jave

- (a) môže mať len abstraktné metódy
- (b) nemôže mať metódy
- (c) nemusí mať len abstraktné metódy
- (d) nemôže mať inicializované atribúty
- (e) nemôže mať prekonávajúce metódy

9. (2b) 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 deklarovať alebo ošetrovať výnimku typu MyException
- (b) musí deklarovať že vyhadzuje výnimku typu MyException
- (c) je korektná
- (d) musí vyhadzovať výnimku typu MyException
- (e) musí ošetrovať výnimku typu MyException

10. (1b) Nech n je atribút triedy. Pole r je v jednej z metód tej istej triedy definované takto:

```
Integer r[] = new Integer[n];
```

Táto definícia je v Java

- (a) korektná jedine ak je atribút n statický
- (b) nekorektná
- (c) korektná jedine ak je atribút n finálny
- (d) korektná
- (e) korektná jedine ak je atribút n synchronizovaný

11. (1b) Zapuzdrenie sa v triedach v Java

- (a) nedá riadiť priamo
- (b) riadi modifikátormi prístupu
- (c) riadi modifikátorom **static**
- (d) riadi abstraktnými metódami
- (e) čiastočne riadi synchronizáciou metód

12. (1b) Pri rozhodovaní o použití dedenia medzi dvomi triedami treba brať do úvahy predovšetkým

- (a) vzťahy medzi reálnymi objektmi, ktoré triedy reprezentujú
- (b) pohľad predpokladaného klientskeho kódu
- (c) jeho vplyv na efektívnosť kódu
- (d) matematickú abstrakciu objektov, ktoré triedy reprezentujú
- (e) pohľad akéhokoľvek možného klientskeho kódu

15 b

1 b

2 a

3 d

4 b

5 e

6 d

7 c

8 c

9 a

10 d

11 b

12 e