

Objektovo-orientované programovanie

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

Test – 2. apríl 2014

A

| | |
|-------------|-----------------|
| Priezvisko: | tlačeným písmom |
| Meno: | |

| | |
|----|--|
| 1b | |
| 2b | |

| | |
|----|--|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |
| 8 | |
| 9 | |
| 10 | |
| 11 | |
| 12 | |

Test trvá 30 minút.

Odpovede na otázky vpíšete do tabuľky. Hodnotia sa len odpovede v tabuľke. Otázky sú s výberom odpovede, z ktorých je len jedna možnosť správna.

V prípade opravy jasne vyznačte odpoveď, ktorá platí. Každá správna odpoveď má hodnotu vyznačenú v otázke. Nesprávna, nejednoznačná alebo neúplná odpoveď má hodnotu 0 bodov. Postup riešenia sa nehodnotí.

Poškodený list nebude uznaný.

1. (1 b) V programe v Jave je daná trieda `Objednavka`. Výraz `Objednavka.class` predstavuje referenciu na

- (a) názov triedy `Objednavka` typu `String`
- (b) objekt, ktorý reprezentuje triedu `Objednavka`
- (c) súbor, ktorý reprezentuje triedu `Objednavka`
- (d) inštanciu triedy `Objednavka`
- (e) atribút triedy `Objednavka`

2. (1 b) Zavedenie viacerých balíkov v Jave môže spôsobiť

- (a) zvýšenie veľkosti class súborov tried, ktoré používajú prvky príslušného balíka
- (b) zvýšenie veľkosti výsledného JAR súboru
- (c) neriešiteľnú kolíziu názvov zavedených prvkov
- (d) riešiteľnú kolíziu názvov zavedených prvkov
- (e) spomalenie programu

3. (2 b) K triedam na obr. 1 je daný nasledujúci kód v Jave:

```
((B) o1).m();
```

Tento kód

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

4. (1 b) Typ výnimky v Jave je určený jej

- (a) triedou
- (b) názvom
- (c) identifikačným číslom
- (d) krátkym opisom
- (e) metódami

```
class P {
    public void m() { System.out.print("p."); }
    public static void s() {
        System.out.print("sp.");
    }
}
class A extends P {
    public void m() { System.out.print("a."); }
    public static void s() {
        System.out.print("sa.");
    }
}
class B extends P {
    public void m() { System.out.print("b."); }
    public static void s() {
        System.out.print("sb.");
    }
}
class M {
    public static void op(P o) {
        o.m();
        o.s();
    }
    public static void main(String[] args) {
        A o1 = new A();
        P o2 = new B();

        M.op(o1);
        M.op((P)o1);
        M.op(o2);
        M.op((P)o2);
    }
}
```

Obr. 1: Kód pre otázky 3 a 7.

5. (1 b) Koľko inštancií môže mať anonymná trieda?

- (a) musí mať aspoň jednu, ale môže ich mať viac
- (b) nemusí mať ani jednu, ale môže ich mať viac
- (c) má vždy práve jednu
- (d) nemusí mať ani jednu, ale môže mať jednu
- (e) ani jednu

6. (1 b) Modifikátory prístupu v Jave umožňujú

- (a) redukovať závislosti v kóde
- (b) regulovať, či iní môžu vidieť kód a zasahovať doň
- (c) regulovať, či iní môžu vidieť kód
- (d) redukovať počet atribútov
- (e) redukovať počet metód

7. (2 b) Daný je kód v Jave na obr. 1. Čo sa vypíše vykonaním triedy `M`?

- (a) a.sp.p.sp.b.sp.p.sp.
- (b) a.sa.p.sp.b.sb.p.sp.
- (c) a.sa.a.sp.b.sb.b.sp.
- (d) a.sp.a.sp.b.sp.b.sp.
- (e) a.sa.a.sa.b.sb.b.sb.

8. (1 b) Dedenie v objektovo-orientovanom programovaní treba aplikovať

- (a) čím menej
- (b) berúc do úvahy implicitné správanie a štruktúru
- (c) len pri anonymných triedach
- (d) berúc do úvahy implicitné predpoklady a dôsledky
- (e) čím viac

9. (1 b) V objektovo-orientovanom programe hlavná funkcionálnosť typicky

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

10. (1 b) Inštancia abstraktnej triedy v Jave

- (a) je abstraktná
- (b) je statická
- (c) je generická
- (d) je polymorfná
- (e) nejestvuje

11. (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

12. (2 b) Pokus o preklad a vykonanie nasledujúcej triedy v Jave:

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

- (a) prekladáč hlási chybu pre neznámu veľkosť poľa `a`
- (b) prebehne korektne
- (c) pri vykonávaní vznikne výnimka pre neznámu veľkosť poľa `a`
- (d) prekladáč hlási chybu pre nekonzistenciu typov
- (e) pri vykonávaní vznikne výnimka pre nekonzistenciu typov

15 b

1 b

2 d

3 e

4 a

5 a – bude akceptovaná aj odpoveď c

6 a

7 d

8 d

9 a

10 e

11 c

12 b