

# Objektovo-orientované programovanie

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

Skúška – 29. máj 2014

B

(vyplňte tlačенým písmom)

Priezvisko:

Meno:

1 b	
2 b	
3 b	

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

Skúška trvá 70 minút.

Odpovede na otázky 1–11 vpíšte do tabuľky. Pri týchto otázkach sa hodnotia len odpovede v tabuľke (bez postupu). Odpoveď musí byť jednoznačná a čitateľná, inak má hodnotu 0 bodov. V otázkach s ponúknutými odpoveďami je len jedna možnosť správna – do tabuľky píšete len písmeno, ktorým je označená odpoveď, ktorú vyberáte.

Odpoveď na otázku 12 píšete výlučne na list, na ktorom sa nachádza jej znenie.

Poškodený list nebude uznaný.

1. (2 b) Pre ktorý návrhový vzor je charakteristická notifikácia určitých objektov pri zmene stavu iného objektu?

2. (2 b) V programe je každý druh geometrického útvaru reprezentovaný triedou. Rozmery útvarov sú reprezentované atribútmi a ich čítanie a zmena zabezpečené prostredníctvom metód. Je vhodné odvodiť triedu, ktorá reprezentuje štvorec, od triedy, ktorá reprezentuje obdĺžnik?

- (a) nie, lebo to porušuje Liskovej princíp substitúcie
- (b) áno
- (c) nie, lebo to porušuje princíp otvorenosti a uzavretosti
- (d) nie, lebo sa tým zbytočne zväčší program
- (e) áno, ale prekonaním metód musí byť zabezpečené, aby sa pri štvorcovej výška menila zároveň so šírkou

3. (1 b) Viacnásobné dedenie v jazyku C++

- (a) slúži na zníženie počtu tried
- (b) je to isté, čo opakované dedenie
- (c) nahrádza mechanizmus friend
- (d) predstavuje alternatívu k virtuálnym funkciám
- (e) môže prispieť k lepšiemu oddeleniu záležitostí

4. (3 b) Súčasťou grafického používateľského rozhrania počítačovej hry je aj tlačidlo t (objekt typu JButton), v súvislosti s ktorým sa v hre vyskytuje nasledujúci kód v Jave:

```
...
class DecreaseEnergy implements ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (player.hasShield())
            player.setEnergy(player.getEnergy() - 1);
        else
            player.setEnergy(player.getEnergy() - 2);
    }
}
...
```

```
t.addActionListener(new DecreaseEnergy());
...
```

Primárny problém tohto kódu z hľadiska objektovo-orientovaného návrhu je to, že

- (a) nebolo použité zapuzdrenie
- (b) nebol použitý polymorfizmus
- (c) aplikačná logika figuruje v používateľskom rozhraní
- (d) neobsahuje komentár
- (e) prijímač nebol realizovaný prostredníctvom anonimnej triedy

5. (3 b) Daný je nasledujúci program v Jave:

```
class A {
    private static int a = 'a', b = 'b';
    public static void m() {
        if (a == 'a') {
            a = 'b';
            b = 'a';
        }
        else {
            a = 'a';
            b = 'b';
        }
        if (a == b)
            System.out.println("=");
    }
    public static void main(String[] args) {
        A a = new A();
        (new B(a)).start();
        (new B(a)).start();
    }
}
class B extends Thread {
    A a;
    public B(A a) {
        this.a = a;
    }
    public void run() {
        for (int i = 1; i < 100000; i++)
            a.m();
    }
}
```

Ktoré metódy musia byť synchronizované, aby sa nikdy nevypísal znak =?

6. (1 b) V jazyku C++ prekonávanie sa použitím špeciálneho kľúčového slova

- (a) musí v prípade, že nie je potrebné, deaktivovať
- (b) musí v prípade potreby aktivovať
- (c) musí v prípade potreby aktivovať, a následne sa môže deaktivovať
- (d) musí v prípade potreby aktivovať, a následne sa musí aj deaktivovať
- (e) môže v prípade, že nie je potrebné, deaktivovať

7. (1 b) Bodový prierez v jazyku AspectJ

- (a) rozdeľuje program v stanovených bodoch
- (b) ovplyvňuje program v stanovených miestach
- (c) okrem iného, mení hodnoty parametrov metód
- (d) zachytáva miesta, v ktorých program môže byť ovplyvnený
- (e) pridáva nové prvky do jestvujúcich tried

8. (2 b) Daná je trieda T v Jave:

```
class T {  
    void m() {  
        try {  
            new C().f();  
        } catch (MyException e) {  
            ...  
        }  
    }  
}
```

Z tohto kódu možno usúdiť, že kód naznačený tromi bodkami

- (a) sa vykoná, ak metóda f() vyhodí výnimku typu MyException
- (b) sa vykoná, ak metóda f() nevyhodí výnimku typu MyException
- (c) má zabezpečiť zachytenie výnimky typu MyException
- (d) má zabezpečiť zachytenie a spracovanie výnimky typu MyException
- (e) sa vykoná vždy

9. (1 b) V jazyku C# zástupca (delegate) reprezentuje

- (a) hocijakú funkciu
- (b) vlastnosť zodpovedajúceho typu
- (c) funkciu zodpovedajúcich typov parametrov a návratovej hodnoty
- (d) hocijakú vlastnosť
- (e) ukazovateľ na objekt

10. (1 b) Je možné v jazyku AspectJ zabezpečiť vykonávanie metód, pri ktorých je to potrebné, prostredníctvom odosielacej nite rámca Swing?

- (a) nie, lebo ich nie je možné posunúť na vykonávanie do inej nite
- (b) nie, lebo ich nie je možné jednoznačne špecifikovať
- (c) áno, ale je potrebné vykonať isté zásahy do volaní týchto metód
- (d) áno, jednoduchým zahrnutím príslušného aspektu do kompilácie
- (e) áno, ale je potrebné vykonať isté zásahy do tiel týchto metód

11. (3 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
interface I {  
    void m();  
}  
abstract class K implements I {  
    public void m() {  
        System.out.print("K");  
    }  
}  
class L extends K {  
    public void m() {  
        super.m();  
        System.out.print("L");  
    }  
}  
class M extends L {  
    public void m() {  
        super.m();  
        System.out.print("M");  
    }  
}  
class C {  
    public static void exe(I... a) {  
        for (I e : a)  
            e.m();  
    }  
    public static void main(String[] args) {  
        I a = new M();  
        M b = new M();  
        K c = new L();  
        I d = new L();  
  
        exe(a, (L)b, c, d);  
    }  
}
```

(vypláte tlačným písmom)

**Priezvisko:**

**Meno:**

**12. (10 b)** Virtuálny priestor pozostáva z poprepájaných buniek. Bunka môže byť nečleniteľná alebo členiteľná. Členiteľná bunka môže obsahovať ďalšie nečleniteľné a členiteľné bunky. Bunky môžu byť prepojené bez ohľadu na to, či sú nečleniteľné alebo členiteľné. Pri každej bunke je možné získať zoznam buniek, do ktorých možno z tejto bunky vstúpiť, čo sú bunky, s ktorými je daná bunka priamo prepojená, a – v prípade členiteľnej bunky – bunky, z ktorých daná bunka bezprostredne pozostáva (prvá úroveň).

Nakreslite diagram tried s najvýznamnejšími vzťahmi, operáciami a atribútmi, ktoré vyplývajú z uvedeného opisu vnútorného modelu programu (GUI nie je predmetom otázky). Napíšte zodpovedajúci kód v Jave vrátane (vykonštruovaného) príkladu použitia, v ktorom vytvoríte príslušné objekty a spustíte ich interakciu. Aplikujte adekvátne mechanizmy objektovo-orientovaného programovania. Ak je to vhodné, aplikujte niektorý z návrhových vzorov a vysvetlite, čo sa ním dosahuje.

1 Observer

2 a

3 e

4 c

5 A.m() (akceptovateľné aj len m())

6 b

7 d

8 a

9 c

10 d

11 KLMKLMKLL

30