Objektovo-orientované programovanie **B**

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

Skúška – 11. jún 2013

(vyplňte tlačeným písmom)

Priezvisko:

Meno:

1 b	
2 b	
3 b	

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

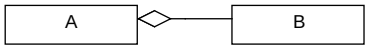
Skúška trvá 75 minút.
Odpovede na otázky 1–12 vpíšte do tabuľky. Pri týchto otázkach sa hodnotia len odpovede v tabuľke (bez postupu). Odpoveď musí byť jednoznačná a čitateľná, inak má hodnotu 0 bodov. V otázkach s ponúknutými odpoveďami je len jedna možnosť správna – do tabuľky píšte len písmeno, ktorým je označená odpoveď, ktorú vyberáte.
Odpoveď na otázku 13 píšte výlučne na list, na ktorom sa nachádza jej znenie.
Poškodený list nebude uznaný.

1. (2 b) Daný je nasledujúci kód v Jave:

```
interface A {
    void m();
}
class B implements A {
    List<A> a;
    public void m() {
        ...
    }
    ...
}
class C implements A {
    public void m() {
        ...
    }
}
```

Predstavuje tento kód určitý návrhový vzor? Ak áno, tak ktorý?

2. (2 b) Daný je nasledujúci diagram v jazyku UML:



Znázornený vzťah sa na úrovni implementácie v Jave prejaví tak, že

(a) trieda A bude obsahovať triedu B
(b) trieda B bude mať atribút typu A
(c) trieda B bude obsahovať triedu A
(d) trieda A bude mať atribút typu B
(e) trieda A bude volať metódy triedy B

3. (1 b) V jazyku C# zástupcovi (delegate) možno priradiť výlučne

(a) funkciu alebo atribút
(b) funkciu
(c) objekt
(d) vlastnosť
(e) hodnotu

4. (3 b) Daný je nasledujúci kód v Jave:

```
interface I {
    **1**
}
class A {
    **2** f(final int i) {
        return new **3**() {
            public int m(int j) {
                return i * j;
            }
        };
    }
}
class B {
    public static void main(String[] args) {
        System.out.println(new A().f(2).m(5));
    }
}
```

Ktoré fragmenty kódu treba v tomto programe doplniť, aby bol funkčný?

(a) `**1**`: **int** m(**int** j); `**2**`: I `**3**`: Object
(b) `**1**`: **int** m(**int** j); `**2**`: I `**3**`: I
(c) `**1**`: I f(**final int** i); `**2**`: I `**3**`: I
(d) `**1**`: **int** m(**int** j); `**2**`: Object `**3**`: I
(e) `**1**`: Object f(**final int** i); `**2**`: Object `**3**`: I

5. (1 b) V jazyku AspectJ

(a) videnie sa vykonáva v kontexte zachytených bodov spájania
(b) bodový prierez sa vykonáva v kontexte zachytených bodov spájania
(c) bodové prierezy zasahujú do základného programu
(d) zachytený bod spájania sa vykonáva v kontexte bodového prierezu
(e) body spájania zasahujú do základného programu

6. (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:

```
t.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (player.hasShield())
            player.setEnergy(player.getEnergy() − 1);
        else
            player.setEnergy(player.getEnergy() − 2);
    }
});
```

Je tento kód v poriadku?

(a) nie, lebo metóda actionPerformed() nie je synchronizovaná
(b) nie, lebo objekt anonymnej triedy nie je vytvorený správne
(c) nie, lebo zasahuje do aplikačnej logiky
(d) nie, lebo sa nevyužíva objekt e
(e) áno, syntakticky a návrhovo

7. (2 b) Dodržanie princípu otvorenosti a uzavretosti sa v OOP dosahuje predovšetkým použitím

(a) polymorfizmu
(b) agregácie
(c) synchronizácie
(d) alternácie
(e) zapuzdrenia

8. (2 b) Java podporuje perzistenciu prostredníctvom

(a) modularizácie
(b) serializácie
(c) agregácie
(d) synchronizácie
(e) radializácie

9. (2 b) V rámci Swing udalosti, ktoré prijímač (listener) zachytáva, sú dané

(a) registráciou rozhrania prijímača pre tieto udalosti
(b) registráciou triedy prijímača pre tieto udalosti
(c) registráciou metódy prijímača pre tieto udalosti
(d) registráciou atribútu prijímača pre tieto udalosti
(e) registráciou objektu prijímača pre tieto udalosti

10. (1 b) Zabránenie vzniku viacerých kópií toho istého atribútu sa pri viacnásobnom dedení v jazyku C++ dosahuje použitím

(a) virtuálnych funkcií
(b) virtuálnych atribútov
(c) čisto virtuálnych funkcií
(d) virtuálneho dedenia
(e) virtuálnych tried

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

```
interface A {
    void f();
}
abstract class X implements A {
    public void f() {
        System.out.print("x");
    }
}
class Y extends X {
    public void f() {
        super.f();
        System.out.print("y");
    }
}
class Z implements A {
    public void f() {
        System.out.print("z");
    }
}
class Q {
    public void m(A... a) {
        for (A e : a)
            e.f();
    }
}
class C {
    public static void main(String[] args) {
        A a = new Y();
        A b = new Z();
        Y c = new Y();
        X d = new Y();
        Z e = new Z();
        A f = new Z();
        Q q = new Q();
        q.m(a, b, c, d, e, f, (X)new Y(), (A)new Z(),
            new Y(), new Z());
    }
}
```

12. (3 b) Daný je nasledujúci program v Jave:

```
class A extends Thread {
    C c;
    public A(C c) {
        this.c = c;
    }
    public void run() {
        for (int i = 0; i < 9999; i++)
            c.a();
    }
}
class B extends Thread {
    C c;
    public B(C c) {
        this.c = c;
    }
    public void run() {
        for (int i = 0; i < 9999; i++)
            c.b();
    }
}
class C {
    private char a = 'a', b = 'b';
    public synchronized void a() {
        if (a != b)
            System.out.println("a");
        a = 'a';
        b = 'a';
    }
    public void b() {
        if (a != b)
            System.out.println("b");
        a = 'b';
        b = 'b';
    }
    public static void main(String[] args) {
        C c = new C();
        new A(c).start();
        new B(c).start();
    }
}
```

Výstupom tohto programu

(a) nebude nič
(b) bude tisíckrát znak a
(c) nebude nič alebo budú znaky `a` a `b` v nepravidelnom počte a striedaní
(d) budú tisíckrát znaky `a` a `b` v nepravidelnom striedaní
(e) bude tisíckrát znak b

OOP – skúška – 11. jún 2013

(vyplňte tlačeným písmom)

Priezvisko:

Meno:

13. (10 b) V počítačovej hre hlavná postava, ktorú riadi hráč, môže mať u seba magické veci rôznych druhov. Účinnosť týchto vecí závisí od energie postavy: čarovný prútik je účinný, len ak postava má viac než 30% energie, čarovný prsteň je účinný, len ak postava má viac než 50% energie, kým čarovný medailón je účinný, len ak postava má viac než 70% energie. Hlavná postava môže mať u seba viac vecí a to aj rovnakého druhu. Plánované je rozšírenie hry o nové druhy vecí.

Nakreslite diagram tried s najvýznamnejšími vzťahmi, operáciami a atribútmi, ktoré vyplývajú z uvedeného opisu vnútorného modelu hry (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.

Objektovo-orientované programovanie **B**
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška – 11. jún 2013

1 Composite

2 d

3 b

4 b

5 a

6 c

7 a

8 b

9 e

10 d

11 xyzxyxyzzxyzxyz

12 c

35