Objektovo-orientované programovanie **A**

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

Opravná skúška – 1. júl 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.

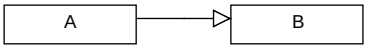
1. (3 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
abstract class A {
    public static int i;
    public void m() {
        System.out.print("a");
    }
}
class B extends A {
    public void m() {
        System.out.print("b");
        i++;
    }
}
class C extends B {
    public void m() {
        System.out.print("c");
        i--;
    }
}
class M {
    public static void main(String[] args) {
        new B().m();
        new C().m();
        ((B)new C()).m();
        ((A)new B()).m();
        ((A)new C()).m();
        System.out.print(A.i);
    }
}
```

2. (1 b) V jazyku C# sa pri práci s vlastnosťami (properties) prístupove a nastavovacie funkcie (set/get) volajú

(a) implicitne
(b) implicitne alebo explicitne
(c) explicitne
(d) implicitne pri modifikácii, a explicitne pri prístupe
(e) implicitne pri prístupe, a explicitne pri modifikácii

3. (2 b) Daný je nasledujúci diagram v jazyku UML:



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

(a) trieda B bude dediť od triedy A
(b) trieda A bude obsahovať triedu B
(c) trieda A bude dediť od triedy B
(d) trieda B bude obsahovať triedu A
(e) trieda A bude vplývať na triedu B

4. (2 b) Daný je nasledujúci kód v Jave:

```
interface A {
    void m(X x);
}
interface X {
    ...
}
class C implements A {
    public void m(X x) {
        x.op(this);
    }
}
```

Pre ktorý návrhový vzor je tento kód charakteristický?

5. (1 b) V jazyku C++ prekonávanie sa použitím špeciálneho kľúčového slova

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

6. (2 b) Podľa Liskovej princípu substitúcie

(a) pri použití dedenia treba dbať na kód, ktorým je implementované
(b) pri použití dedenia treba dbať na klientsky kód
(c) treba čím viac používať dedenie
(d) pri použití dedenia treba dbať na reálne alebo matematické objekty
(e) treba čím menej používať dedenie

7. (2 b) Metóda f() triedy A vyhadzuje výnimku MyException. Daná je trieda B:

```
class B {
    void m() {
        new A().f();
    }
}
```

Metóda m() triedy B

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

8. (2 b) Okno v rámci Swing sa dá vytvoriť zavolaním

(a) metódy newWindow() triedy JFrame
(b) statickej metódy EventQueue.newWindow()
(c) konštruktora triedy SwingWindow
(d) konštruktora triedy JFrame
(e) statickej metódy SwingUtils.newWindow()

9. (1 b) V implementácii vzoru Observer v jazyku AspectJ

(a) aplikačná logika je vyčlenená do aspektu, kým logika vzoru zostáva v triedach
(b) aplikačná logika a logika vzoru sú zmiešané v triedach a aspektoch
(c) aplikačná logika a logika vzoru zostávajú v triedach
(d) logika vzoru je vyčlenená do aspektu, kým aplikačná logika zostáva v triedach
(e) aplikačná logika a logika vzoru sú vyčlenené do aspektu

10. (3 b) Hra v Jave je ovládaná stlačením zodpovedajúcich tlačidiel takto:

```
switch (o)
case 'a':
    if (player1.getEnergy() < 10)
        player2.addObject(player1.takeObject());
    else
        player1.addEnergy(player2.getEnergy());
    break;
case 'b':
...
```

Z hľadiska kvality objektovo-orientovaného návrhu by bolo vhodné

(a) vyčleniť kód jednotlivých prípadov (case) do zodpovedajúcich metód
(b) ponechať všetko tak, ako je
(c) implementovať ovládanie myšou namiesto klávesnicou
(d) použiť polymorfizmus
(e) použiť zapuzdrenie

11. (3 b) Daný je nasledujúci kód v Jave:

```
abstract class A {
}
class X extends A {
}
class Y extends X {
}
class M {
    static void m(Class<? extends A> T, A... o) {
        int i = 0;
        for (A e : o) {
            if (T.isInstance(e))
                i++;
        }
        System.out.print(i);
    }
    public static void main(String[] args) {
        m(Y.class, new A[]{new X(), (X)new Y(),
            (A)new Y()});
    }
}
```

Aká hodnota sa vypíše po jeho vykonaní?

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

```
class A {
    private int i1 = 1, i2 = 1;
    public void m() {
        if (i1 == 1) {
            i1 = 2;
            i2 = 2;
        }
        else {
            i1 = 1;
            i2 = 1;
        }
    }
}
class B implements Runnable {
    A a;
    public B(A a) {
        this.a = a;
    }
    public void run() {
        for (int i = 1; i < 100000; i++)
            a.m();
    }
}
class M {
    public static void main(String[] args) {
        A a = new A();
        new Thread(new B(a)).start();
        new Thread(new B(a)).start();
    }
}
```

Aby hodnota atribútu i1 bola vždy rovnaká ako hodnota atribútu i2

(a) metóda m() musí byť synchronizovaná
(b) metóda run() musí byť synchronizovaná
(c) metóda m() a metóda run() musia byť synchronizované
(d) metóda main() musí byť synchronizovaná
(e) metóda run() a metóda main() musia byť synchronizované

OOP – opravná skúška – 1. júl 2013

(vyplňte tlačeným písmom)

Priezvisko:

Meno:

13. (10 b) V systéme sa okrem iného uchováva zoznam textových položiek (spôsob implementácie zoznamu zvoľte; položky môžu byť obmedzenej veľkosti). Položky je vo všeobecnosti možné filtrovať podľa rôznych kritérií, výsledkom čoho je zoznam položiek, ktoré toto kritérium spĺňajú. Spôsoby filtrovania majú tendenciu pribúdať a to v ľubovoľnej programovej zložitosti.

Nakreslite diagram tried s najvýznamnejšími vzťahmi, operáciami a atribútmi, ktoré vyplývajú z uvedeného opisu vnútorného modelu systému (GUI nie je predmetom otázky). Zahŕňte dva špecifické spôsoby filtrovania. Napíšte zodpovedajúci kód v Jave vrátane (vykonštruovaného) príkladu použitia, v ktorom demonštrujete použitie filtrovania. Aplikujte adekvátne mechanizmy objektovo-orientovaného programovania. Ak je to vhodné, aplikujte niektorý z návrhových vzorov a vysvetlite, čo sa ním dosahuje.

Objektovo-orientované programovanie **A**
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Opravná skúška – 1. júl 2013

1 bccbc-1

2 a

3 c

4 Visitor

5 a

6 b

7 e

8 d

9 d

10 a

11 2

12 a

35