

Objektovo-orientované programovanie

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

Opravná skúška – 24. jún 2014

(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. (1 b) V jazyku C++ virtuálne funkcie slúžia na zabezpečenie

- (a) polymorfizmu
- (b) virtuálnej reality
- (c) fiktívnej funkcionality
- (d) preťaženia
- (e) generickosti

2. (1 b) Mechanizmus friend v jazyku C++

- (a) umožňuje poskytnúť prístup k neverejným prvkom triedy iným triedam
- (b) nejestvuje
- (c) sa používa na označenie user-friendly funkcionality
- (d) slúži na zapínanie a vypínanie polymorfizmu
- (e) umožňuje poskytnúť prístup k neverejným prvkom triedy cudzím funkciám

3. (1 b) Vlastnosti (properties) v jazyku C# zodpovedajú v Jave

- (a) prístupovým a nastavovacím metódam (set/get)
- (b) atribútom vo všeobecnosti
- (c) atribútom s pridrúženými prístupovými a nastavovacími metódami (set/get)
- (d) objektom anonýmnych tried
- (e) metódam vo všeobecnosti

4. (1 b) V jazyku AspectJ je možné kódom mimo metódy (bez zásahu do jej kódu a volaní)

- (a) len znemožniť alebo povoliť jej vykonanie, vrátane jej opakovaného vykonania
- (b) nahradiť ju úplne iným kódom vrátane vynechania jej vykonania
- (c) len zmeniť návratovú hodnotu
- (d) nahradiť ju len dostatočne podobným kódom
- (e) nahradiť ju úplne iným kódom, ale nie je možné vynechať jej vykonanie

5. (1 b) V jazyku C# sa kód spracovania udalosti (handler) akou je napríklad kliknutie zabezpečuje

- (a) anonymnou triedou
- (b) anonymnou metódou
- (c) anonymným objektom
- (d) anonymnou vlastnosťou
- (e) anonymným atribútom

6. (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?

7. (2 b) Podľa Liskovej princípu substitúcie

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

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

9. (3 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
interface I {
    void m();
}
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 K();
        K b = new L();
        I c = new M();
        I d = new L();
        M e = new M();

        exe(a, (I)b, c, d, e);
    }
}
```

10. (3 b) Daný je nasledujúci program v Jave:

```
class A {
    private static int a = 'a', b = 'b';
    public void m() {
        if (a == 'a') {
            a = 'b';
            b = 'a';
        }
        else {
            a = 'a';
            b = 'b';
        }
        if (a == b)
            System.out.println("=");
    }
}
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 synchronized void main(String[] args) {
        A a = new A();
        new Thread(new B(a)).start();
        new Thread(new B(a)).start();
        new Thread(new B(a)).start();
        new Thread(new B(a)).start();
        new Thread(new B(a)).start();
    }
}
```

Zaručí uvedenie modifikátora **synchronized** pri metóde `main()`, že sa nevypíše znak =?

- (a) áno
- (b) nie, ale zníži jeho výskyt
- (c) áno, ale len ak aj metóda `m()` bude synchronizovaná
- (d) nie, to nebude mať žiaden vplyv
- (e) áno, ale len ak aj metóda `m()` bude synchronizovaná a statická

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

(vypláte tlačným písmom)

Priezvisko:

Meno:

12. (10 b) V urbanistickom simulačnom programe sa vo vzťahu k mestu vyskytujú rôzne dekorácie. Zatiaľ sú to farebné svetlá a fontány, ale v budúcnosti pribudnú ďalšie druhy. V meste sa šetrí energiou a dekorácie obyvateľom naznačujú, kedy je to potrebné. Tak farebné svetlá svietia modro, keď je spotreba mesta pod 50% dlhodobého priemeru, žltó, keď je spotreba mesta medzi 50% a 70% dlhodobého priemeru, a červeno, keď je spotreba mesta nas 70% dlhodobého priemeru. Fontány sú v prevádzke len keď je spotreba mesta pod 60% dlhodobého priemeru.

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.

Objektovo-orientované programovanie

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

Opravná skúška – 24. jún 2014

1 a

2 e

3 c

4 b

5 b

6 Visitor

7 c

8 a

9 KKLKLMKLLKLM

10 d

11 b

30