

Objektovo-orientované programovanie

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

Skúška – opravný termín – 29. jún 2016

(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. (3 b) Trieda, ktorá reprezentuje špeciálny dokument, je odvodená od triedy, ktorá reprezentuje všeobecný dokument. Metóda na pridanie signatára je v špeciálnom dokumente prekonaná tak, že povoľuje pridanie signatára iba z úrovne vyššieho manažmentu, kým vo všeobecnom dokumente na pozícii signatára nezáleží. Týmto sa predpoklady a dôsledky pôvodnej metódy zoslabujú, zosilňujú alebo sa nemenia? Je týmto dodržaný Liskovej princíp substitúcie (LSP)?

Odpovedzte vo forme: *predpoklady* / *dôsledky* / *LSP*. Položky *predpoklady* a *dôsledky* nahraďte jednou z možností *zoslabujú sa*, *zosilňujú sa* alebo *nemenia sa*. Položku *LSP* nahraďte jednou z možností *dodržaný* alebo *nedodržaný*.

2. (1 b) V jazyku C# sa kód spracovania udalosti (handler) akou je napríklad kliknutie zabezpečuje

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

3. (1 b) V jazyku AspectJ je možné kódom mimo metódy (bez zásahu do jej kódu a volaní)

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

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

```
class C extends Thread {  
    private static int i = 0;  
    private static int j = 1;  
    private static int k = 0;  
  
    public void ij() {  
        int b = i;  
        i = j;  
        j = b;
```

```
    }  
    public void ki() {  
        k = i;  
    }  
    public void kj() {  
        k = j;  
    }  
    public void eq() {  
        if (i == j)  
            System.out.println("=");  
    }  
}  
class M {  
    public static void main(String[] args) {  
        C o = new C();  
  
        new Thread() {  
            public void run(){  
                for (int i = 1; i < 100000; i++)  
                    o.ij();  
            }  
        }.start();  
  
        new Thread() {  
            public void run(){  
                for (int i = 1; i < 100000; i++)  
                    o.ki();  
            }  
        }.start();  
  
        new Thread() {  
            public void run(){  
                for (int i = 1; i < 100000; i++)  
                    o.kj();  
            }  
        }.start();  
  
        new Thread() {  
            public void run(){  
                for (int i = 1; i < 100000; i++)  
                    o.eq();  
            }  
        }.start();  
    }  
}
```

Ktoré metódy musia byť synchronizované, aby sa nikdy nevy-písal znak =?

5. (1 b) Na rozdiel od Javy jazyk C++

- (a) má explicitné konštruktory
- (b) automaticky ruší nereferencované objekty
- (c) má explicitné deštruktory
- (d) automaticky vytvára objekty
- (e) má implicitné konštruktory

6. (1 b) Analogický mechanizmus k šablónam (templates) v jazyku C++ v Jave

- (a) je generickosť
- (b) nejestvuje
- (c) sú rozhrania
- (d) sú anonymné triedy
- (e) je RTTI

7. (2 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
class E extends Exception {}

class A {
    public void a(int i) throws E {
        if (i > 0)
            System.out.print("A");
        else
            throw new E();
    }
    public void op(int i) throws E {
        try {
            a(i);
        } catch (E e) {
            throw e;
        } finally {
            System.out.print("F");
        }
        System.out.print("O");
    }
    public static void main(String[] args) {
        try {
            new A().op(1);
            new A().op(-1);
        } catch (E e) {
            System.out.print("E");
        }
    }
}
```

8. (3 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
class A {
    public void x() {
        System.out.print("Ax");
    }
    public static void y() {
        System.out.print("Ay");
    }
}
class B extends A {
    public void x() {
        System.out.print("Bx");
    }
    public static void y() {
        System.out.print("By");
    }
}
class C extends B {
    public void x() {
        super.x();
        System.out.print("Cx");
    }
    public static void y() {
        B.y();
        System.out.print("Cy");
    }
}
class M {
    public static void main(String[] args) {
        A o1 = new B();
        B o2 = new B();
        B o3 = new C();
        A o4 = new A();
        C o5 = new C();

        o1.x();
        o1.y();

        o2.x();
        o2.y();
    }
}
```

```
((C)o3).x();
((C)o3).y();

o4.x();
o4.y();

((A)o5).x();
((A)o5).y();
}
```

9. (2 b) Niektoré prvky vyskytujúce sa v danom softvérovom riešení môžu byť zložené. Ku všetkým prvkom je však potrebné zabezpečiť jednotný prístup. Ktorý návrhový vzor by ste použili?

- (a) Observer
- (b) Composite
- (c) MVC
- (d) Visitor
- (e) Strategy

10. (3 b) Hra v Jave obsahuje nasledujúci kód:

```
class Vila {
    private int energia;
    private int zivoty;
    ...
    public void energiaNaZivoty() {
        ziskaneZivoty = energia/100;
        zivoty += ziskaneZivoty;
        energia -= ziskaneZivoty * 100;
        HlavneOkno.okno.pocetZivotov.setText(
            Integer.toString(zivoty));
    }
    ...
}
```

Hlavný problém v tomto kóde z hľadiska flexibility objektovo-orientovaného návrhu je to, že

- (a) metóda mení dva atribúty
- (b) kód na prepočítanie energie na životy nie je súčasťou triedy okna hry
- (c) atribúty sú **private** a odvodeným triedam nebudú prístupné
- (d) kód na prepočítanie energie na životy nie je súčasťou zodpovedajúceho prijímača
- (e) vnútorná logika sa mieša s používateľským rozhraním

11. (1 b) Java podporuje perzistenciu prostredníctvom

- (a) radialisácie
- (b) agregácie
- (c) serializácie
- (d) modularizácie
- (e) synchronizácie

(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, žltá, keď je spotreba mesta medzi 50% a 70% dlhodobého priemeru, a červená, keď je spotreba mesta nad 70% dlhodobého priemeru. Fontány sú v prevádzke len keď je spotreba mesta pod 60% dlhodobého priemeru. Implementácia spôsobu počítania dlhodobého priemeru spotreby mesta nie je súčasťou úlohy.

Navrhňte a implementujte v Jave zodpovedajúce objektovo-orientované riešenie. Základný návrh predložte vo forme diagramu tried s najvýznamnejšími vzťahmi, operáciami a atribútmi.

Aplikujte adekvátne mechanizmy objektovo-orientovaného programovania za účelom zabezpečenia flexibilného návrhu. Ak je to vhodné, aplikujte niektorý z návrhových vzorov. Identifikujte explicitne, čím sú modelované a implementované roly aplikovaného vzoru a vysvetlite, čo sa týmto vzorom dosahuje.

Sústredte sa na aplikačnú logiku – GUI nie je predmetom otázky. Poskytnite príklad použitia, v ktorom vytvoríte príslušné objekty a spustíte ich interakciu.

Odpoveď bude hodnotená podľa nasledujúceho kľúča:

- zabezpečenie základnej funkčnosti – 4 b
- kvalita a flexibilita objektovo-orientovaného návrhu – 6 b

Objektovo-orientované programovanie

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

Skúška – opravný termín – 29. jún 2016

30

1 zosilňujú sa / nemenia sa / nedodržaný

2 e

3 d

4 C.ij() a C.eq()

5 c

6 a

7 AFOFE

8 BxAyBxBxBxCxByCyAxAyBxCxAy

9 b

10 e

11 c

Mesto alebo údaje o spotrebe energie ako také sú reprezentované abstraktnou triedou alebo rozhraním a hrajú rolu Subjectu, od ktorej je odvodený jediný konkrétny typ mesta. Dekorácie sú reprezentované abstraktnou triedou alebo rozhraním a hrajú rolu Observera. Od tejto triedy sú odvodené konkrétne druhy dekorácií. Spotreba mesta (napr. denná) je reprezentovaná príslušným atribútom (napr. int). V metóde, ktorá slúži na nastavenie aktuálnej spotreby, sa aktivuje notifikácia všetkých registrovaných dekorácií, ktoré následne upraví svoj stav reprezentovaný príslušným atribútom (napr. enum alebo Boolean).

Otázka bude hodnotená podľa nasledujúceho kľúča:

- zabezpečenie základnej funkčnosti – 4 b
- návrh vzhľadom na princíp otvorenosti a uzavretosti kódu a adekvátne použitie zapuzdrenia – 6 b

Pri hodnotení oboch častí bude zohľadnené poskytnuté vysvetlenie (vrátane príkladu použitia) a zodpovedajúci diagram tried vo výške približne 10–20% hodnotenia príslušnej časti.