Objektovo-orientované programovanie 2018/19
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška – náhradný termín – 28. máj 2019

(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 70 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 vpíšte 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. (1 b) V jazyku C++ možno vytvoriť nový objekt

(a) iba z tried, ktoré neobsahujú virtuálne funkcie
(b) pomocou operátora **new** ako v Jave alebo definovaním premennej typu danej triedy
(c) nie z triedy, ale iba zo šablóny
(d) výlučne pomocou operátora **new** ako v Jave
(e) výlučne definovaním premennej typu danej triedy

2. (1 b) Mechanizmus friend známy z jazyka C++ v Jave

(a) jestvuje pod názvom **static**
(b) jestvuje pod rovnakým názvom
(c) jestvuje pod názvom package access
(d) nejestvuje
(e) jestvuje pod názvom generics

3. (1 b) V jazyku C# (menný priestor – angl. namespace)

(a) v jednom adresári možno definovať iba jeden menný priestor
(b) prvky z rôznych súborov môžu byť súčasťou jedného menného priestoru, ak sa nachádzajú v spoločnom adresári
(c) možno definovať viac menných priestorov v jednom súbore a prvky z rôznych súborov môžu byť súčasťou jedného menného priestoru
(d) prvky z rôznych súborov môžu byť súčasťou jedného menného priestoru, ale nemožno definovať viac menných priestorov v jednom súbore
(e) možno definovať viac menných priestorov v jednom súbore, ale prvky z rôznych súborov nemôžu byť súčasťou jedného menného priestoru

4. (1 b) Java podporuje perzistenciu prostredníctvom

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

5. (1 b) Je možné V Jave pristúpiť k triede definovanej v inom balíku bez importovania tohto balíka?

(a) nie
(b) áno, ale nedá sa od nej dediť
(c) áno
(d) áno, ale trieda musí byť v tom istom súbore
(e) áno, ale trieda musí byť uložená v tom istom adresári

6. (1 b) Na to, aby v jazyku AspectJ bolo možné vykonať kód pred metódou

(a) treba ju označiť anotáciou @before
(b) nie je potrebné do nej zasahovať
(c) pred jej názvom treba použiť kľúčové slovo before
(d) jej názvu musí predchádzať predponá before
(e) treba ju premiestniť do aspektu

7. (2 b) Čo všetko sa vypíše prostredníctvom príkazov System.out.print() po spustení nasledujúceho programu v Jave (po jeho úspešné alebo neúspešné ukončenie)?

```
class E extends Exception {}

class M {
    public void c(char c) throws E {
        if (c == 'a')
            System.out.print("A");
        else
            throw new E();
    }
    public void m(char c) {
        System.out.print("M");

        try {
            c(c);
        } catch (E e) {
            System.out.print("E");
        } finally {
            System.out.print("F ");
        }
    }
    public static void main(String[] args) {
        new M().m('b');
        new M().m('b');
        new M().m('a');
    }
}
```

8. (2 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) použiť polymorfizmus
(c) implementovať ovládanie myšou namiesto klávesnicou
(d) ponechať všetko tak, ako je
(e) použiť zapuzdrenie

9. (2 b) Daný je nasledujúci program v Jave:

```
class A extends Thread {
    C c;
    public A(C c) {
        this.c = c;
    }
    public synchronized 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 synchronized void run() {
        for (int i = 0; i < 9999; i++)
            c.b();
    }
}
class C {
    private char a = 'a', b = 'a';

    public synchronized void a() {
        if (a != b)
            System.out.println("!");
        a = 'a';
        b = 'a';
    }
    public synchronized void b() {
        synchronized(this) {
            if (a != b)
                System.out.println("!");
            a = 'b';
            b = 'b';
        }
    }
    public static synchronized void main(String[] args) {
        C c = new C();
        new A(c).start();
        new B(c).start();
    }
}
```

Pri ktorých metódach je možné odstrániť modifikátor **synchronized**, aby stále bolo zaručené, že sa výkričník nikdy nevypíše (uveďte ich v zápise: `Trieda.metóda()`)?

10. (2 b) V danom softvérovom riešení je potrebné zastrešiť rôzne spôsoby riešenia toho istého problému. Ktorý návrhový vzor by ste použili?

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

11. (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() {
        super.x();
        System.out.print("Bx");
    }
    public static void y() {
        A.y();
        System.out.print("By");
    }
}
class C extends B {
    public void x() {
        System.out.print("Cx");
    }
    public static void y() {
        System.out.print("Cy");
    }
}
class M {
    public static void main(String[] args) {
        C o1 = new C();
        B o2 = new C();
        A o3 = new B();
        A o4 = new B();
        B o5 = new B();

        ((C) o1).x();
        ((C) o1).y();
        System.out.print(" ");

        ((B) o2).x();
        ((B) o2).y();
        System.out.print(" ");

        o3.x();
        o3.y();
        System.out.print(" ");

        o4.x();
        o4.y();
        System.out.print(" ");

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

12. (3 b) Trieda, ktorá reprezentuje špeciálnu hru, je odvodená od triedy, ktorá reprezentuje všeobecnú hru. Metóda na pridanie hráča je v špeciálnej hre prekonaná tak, že povoľuje pridanie hráča s akýmkoľvek počtom bodov, kým vo všeobecnej hre počet bodov musí byť väčší ako nula (možné je aj záporné hodnotenie). Týmto sa predpoklady a dôsledky týchto metód 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*.

(vyplňte tlačeným písmom)

Priezvisko:

Meno:

13. (10 b) V systéme na správu úloh každá úloha má názov, začiatok realizácie (stačí implementovať ako celé číslo: deň od začiatku projektu), predpokladané trvanie (znovu stačí implementovať ako celé číslo: počet dní od začiatku realizácie) a opis. Úloha môže nadväzovať na (jednu) inú úlohu. Okrem iných, na evidované úlohy sú poskytované tieto dve sumarizácie, ktoré sa automaticky aktualizujú pri zmene údajov úloh:

- zoznam názvov úloh so začiatkom realizácie
- zoznam dvojíc úloh, ktoré sú vo vzťahu nadväznosti

Navrhnite a implementujte v Jave zodpovedajúce objektovo-orientované riešenie zohľadňujúce princípy objektovo-orientovaného programovania. Využite pritom najvhodnejší z návrhových vzorov Strategy, Observer, Visitor a Composite.

Základný návrh predložte vo forme náčrtu diagramu tried v UML, ktorý bude obsahovať najvýznamnejšie vzťahy, operácie a atribúty. Zoberte pritom do úvahy návrhový vzor. Viditeľnosť nie je potrebné uvádzať.

V implementácii sa sústreďte na aplikačnú logiku – GUI nie je predmetom otázky. Taktiež, použité algoritmy nemusia byť optimálne.

Identifikujte explicitne prvky, ktorými sú modelované a implementované roly aplikovaného návrhového vzoru, a vysvetlite, prečo ste ho aplikovali. 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 2018/19
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška – náhradný termín – 28. máj 2019
30

1 b

2 d

3 c

4 e

5 c

6 b

7 MEF MEF MAF (akceptovaná je aj odpoveď bez medzier)

8 a

9 A.run(), B.run(), C.b() a C.main()

10 e

11 CxCy CxAyBy AxBxAy AxBxAy AxBxAy

12 zoslabujú sa / nemenia sa / dodržaný

V poslednej otázke mal byť aplikovaný vzor Observer. Predmetom pozorovania mal byť objekt triedy, ktorá slúži na uchovanie úloh (rola Subject), a pozorovateľmi objekty tried, ktoré implementujú sumarizácie (rola Observer).