Objektovo-orientované programovanie 2021/22 **B**

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

Skúška – 1. jún 2022

(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) Tomu, čo sa v Jave implementuje pomocou atribútov s pridruženými prístupovými a nastavovacími metódami (set/get) v jazyku C# zodpovedá mechanizmus

(a) delegate
(b) var
(c) sealed
(d) event
(e) property

2. (1 b) V jazyku C++ možno vytvoriť nový objekt

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

3. (1 b) Analogickým mechanizmom k šablónam (templates) v jazyku C++ v Jave

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

4. (1 b) Delegát v jazyku C# je formou ukazovateľa na

(a) šablónu
(b) funkciu
(c) triedu
(d) objekt
(e) makro

5. (1 b) Je možné len pomocou medzitypových deklarácií (bez použitia RTTI) v jazyku AspectJ ovplyvniť správanie programu?

(a) áno, ale len statických metód
(b) nie
(c) áno
(d) áno, ale nie nastavovanie atribútov
(e) áno, ale len nestatických metód

6. (1 b) Serializácia v Jave slúži na

(a) ochranu údajov
(b) uloženie zdrojového kódu
(c) synchronizáciu údajov
(d) uchovanie údajov
(e) agregáciu údajov

7. (2 b) Daný je nasledujúci program v Jave:

```
class C implements Runnable {
    M m;
    public C(M m) {
        this.m = m;
    }
    public synchronized void run() {
        for (int i = 0; i < 100000; i++) {
            m.a();
            m.c();
        }
    }
}

public class M {
    private char x = 'a', y = 'a';

    public synchronized void a() {
        if (x != y)
            System.out.println("a");
        x = 'a';
        y = 'a';
    }

    public synchronized void b() {
        synchronized(this) {
            if (x != y)
                System.out.println("b");
            x = 'b';
            y = 'b';
        }
    }

    public synchronized void c() {
            synchronized(this) {
                if (x != y)
                    System.out.println("c");
                x = 'c';
                y = 'c';
            }
        }

    public static synchronized void main(String[] args) {
        M c = new M();
        new Thread(new C(c)).start();
        new Thread(new C(c)).start();
    }
}
```

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

8. (2 b) Môžu vznikať rôzne spôsoby realizácie určitej funkčnosti v danom kontexte. Ktorý návrhový vzor by ste použili?

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

9. (2 b) Čo všetko sa vypíše prostredníctvom príkazov System.out.print() po spustení nasledujúceho programu v Jave (po jeho úspešné alebo neúspešné ukončenie)?

```
public class E {
    public void c(int n) throws Exception {
        if (n == 0)
            throw new Exception();
    }

    public void m(int n) {
        try {
            c(n);
        } catch (Exception e) {
            System.out.print("E");
        } finally {
             System.out.print(n);
        }
    }

    public static void main(String[] args) {
        E e = new E();
        e.m(2);
        e.m(0);
        e.m(0);
    }
}
```

10. (2 b) Zabezpečuje vzor MVC oddelenie vnútornej logiky a používateľského rozhrania? Ak áno, ktorá rola tohto vzoru prevažne implementuje vnútornú logiku, a ktorá používateľské rozhranie.

Odpovedzte vo forme: *oddelenie / používateľské rozhranie / vnútorná logika*. Položku *oddelenie* nahraďte jednou z možností *áno* alebo *nie* podľa toho, či si myslíte, že vzor MVC zabezpečuje oddelenie vnútornej logiky a používateľského rozhrania alebo nie. Ak ste odpovedali *áno*, položky *používateľské rozhranie* a *vnútorná logika* nahraďte príslušnými rolami tohto vzoru.

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

```
class C {
    public void f() {
        System.out.print("C");
    }
}
class D extends C {
    public void f() {
      super.f();
        System.out.print("D");
    }
}
class E extends D {
    public void f() {
      super.f();
      System.out.print("E");
    }
}
class F extends E {
    public void f() {
      super.f();
        System.out.print("F");
    }
}
public class M {
    public static void main(String[] args) {
        C o1 = new F();
        F o2 = new F();
        D o3 = new E();
        E o4 = new E();
        C o5 = new D();

        ((C) o1).f();
        System.out.print(" ");

        ((E) o2).f();
        System.out.print(" ");

        ((E) o3).f();
        System.out.print(" ");

        ((D) o4).f();
        System.out.print(" ");

        ((C) o5).f();
    }
}
```

12. (3 b) Trieda, ktorá v hre reprezentuje vesmírnu loď s automatizovaným riadením, je odvodená od triedy, ktorá reprezentuje základnú vesmírnu loď. Metóda na priradenie kapitána lode, ktorej jedným z parametrov je práve objekt, ktorý reprezentuje kapitána, je vo vesmírnej lodi s automatizovaným riadením prekonaná a pripúšťa aj kapitána nižšej kategórie ako pri základných vesmírnych lodiach. Týmto sa predpoklady a dôsledky pôvodnej metódy zoslabujú, zosilňujú alebo nemenia? Je vhodné od základnej vesmírnej lode odvodiť vesmírnu loď s automatizovaným riadením? Je týmto dodržaný Liskovej princíp substitúcie (LSP)?

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

(vyplňte tlačeným písmom)

Priezvisko:

Meno:

13. (10 b) V zjednodušenej počítačovej reprezentácii mesto pozostáva zo stavebných objektov, ktoré predstavujú akékoľvek budovy alebo samostatné celky v rámci týchto budov. Každý takýto samostatný celok sa môže členiť na ďalšie samostatné celky a miestnosti, ktoré sa už ďalej nečlenia.

Používateľ môže vstúpiť do stavebného objektu, pričom systém vypíše zoznam identifikátorov stavebných objektov a miestností, z ktorých tento stavebný objekt pozostáva. V prípade, že používateľ vstúpil do miestnosti, systém vypíše iba identifikátor tejto miestnosti.

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 prvkov nie je potrebné uvádzať.

V implementácii sa sústreďte na aplikačnú logiku – používateľské rozhranie 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 2021/22 **B**
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška – 1. jún 2022
30

1 e

2 c

3 a

4 b

5 b

6 d

7 C.run(), M.b(), M.c(), M.main()

8 a

9 2E0E0

10 áno / View / Model

11 CDEF CDEF CDE CDE CD (akceptovaná je aj odpoveď bez medzier)

12 zoslabujú sa / nemenia sa / áno / dodržaný

13 Vhodný je vzor Composite. Stavebné objekty by boli v role Composite, a miestnosti v role Leaf. Obidva druhy objektov by boli zastrešené rolou Component.