Objektovo-orientované programovanie 2022/23 **A**
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška – 31. máj 2023

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 AspectJ

(a) je možné zachytiť volanie metódy a nechať vykonať kód pred, po alebo namiesto volania
(b) nie je možné zachytiť volanie metódy
(c) je možné zachytiť volanie metódy a nechať vykonať kód iba pred alebo po volaní, ale nie aj namiesto volania
(d) je možné zachytiť volanie metódy a nechať vykonať kód iba namiesto volania, ale nie aj pred alebo po volaní
(e) je možné zachytiť volanie metódy, ale nie aj nechať vykonať kód pred, po, a ani namiesto volania

2. (1 b) Spracovanie výnimky v Jave sa realizuje

(a) blokom kódu spracovania na mieste jej vyhodenia
(b) implementáciou metódy **try**() v triede výnimky
(c) automaticky v metóde main()
(d) implementáciou metódy **catch**() v triede výnimky
(e) blokom kódu spracovania na mieste jej zachytenia

3. (1 b) Ak by ste v jazyku C++ implementovali sčítavanie a odčítavanie farieb, ktoré sú reprezentované inštanciami príslušnej triedy, na umožnenie zápisu týchto operácii pomocou operátorov + a − treba tieto operátory

(a) generalizovať
(b) konkretizovať
(c) virtualizovať
(d) preťažiť
(e) prekonať

4. (1 b) Zahrnutie objektu jednej triedy v objekte inej triedy sa v objektovo-orientovanom programovaní označuje ako

(a) dedenie
(b) agregácia
(c) integrácia
(d) segregácia
(e) generickosť

5. (1 b) Vlastnosti (properties) v jazyku C# umožňujú

(a) ovplyniť kód ako aspekty v jazyku AspectJ
(b) zrýchliť vykonávanie programu
(c) jednoduchšie zabezpečiť zapúzdrenie
(d) vytvárať zástupcov funkcií
(e) zachytiť udalostí

6. (1 b) Výhodou použitia aspektov v jazyku AspectJ je, že

(a) zrychľujú vykonávanie programu
(b) umožňujú oddeliť prepletené záležitostí
(c) automaticky opravujú chyby v programe
(d) zvyšujú bezpečnosť
(e) umožňujú generovať lepšiu dokumentáciu než Javadoc

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

```
class Zero extends Thread {
    C c;
    public Zero(C c) {
        this.c = c;
    }
    public synchronized void run() {
        for (int i = 0; i < 9999; i++)
            c.zero();
    }
}
class One extends Thread {
    C c;
    public One(C c) {
        this.c = c;
    }
    public void run() {
        for (int i = 0; i < 9999; i++)
            c.one();
    }
}
class C {
    private int d1 = 0, d2 = 0;
    public synchronized void zero() {
        if (d1 != d2)
            System.out.println(0);
        d1 = 0;
        d2 = 0;
    }

    public void one() {
        if (d1 != d2)
            System.out.println(1);
        d1 = 1;
        d2 = 1;
    }
    public static void main(String[] args) {
        C c = new C();
        new Zero(c).start();
        new One(c).start();
    }
}
```

Výstupom tohto programu aj pri opakovanom vykonávaní

(a) nebude nič alebo budú nuly a jednotky v nepravidelnom počte a striedaní
(b) bude vždy desaťtisíc núl a jednotiek v nepravidelnom striedaní
(c) bude vždy desaťtisíc jednotiek
(d) nebude nič
(e) bude vždy desaťtisíc núl

8. (2 b) Potrebné je prístupovať rovnakým spôsobom aj k entitám, ktoré sú jednoduché, aj k takým, ktoré sú zložené z iných entít. Ktorý návrhový vzor by ste použili?

(a) Observer
(b) MVC
(c) Strategy
(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)?

```
class E extends Exception {}

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

   public void f(char c) throws E {
      System.out.print("F");

      try {
         m(c);
      } catch (E e) {
         System.out.print("1");
      } finally {
         System.out.print("2");
      }
   }

   public static void main(String[] args) throws E {
      new M().f('a');
      new M().f('b');
      new M().f('b');
   }
}
```

10. (2 b) Vzhľadom na výsledný kód po kompilácii, pre príkazy **import** v Jave a **include** v C++ platí, že:

(a) ani jeden z nich nespôsobuje jeho nárast
(b) obidva príkazy za istých okolností môžu spôsobiť jeho nárast
(c) **include** nespôsobuje jeho nárast, kým **import** áno
(d) **import** nespôsobuje jeho nárast, kým **include** len občas
(e) **import** nespôsobuje jeho nárast, kým **include** áno

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) {
      B o1 = new C();
      A o2 = new B();
      B o3 = new B();
      C o4 = new C();
      A o5 = new B();

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

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

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

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

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

12. (3 b) Trieda, ktorá reprezentuje štvormiestne vozidlo, poskytuje metódu **void** umiestni(**int** miesto, Osoba osoba), ktorá umiestni danú osobu na dané miesto označené číslom 1–4. Trieda, ktorá reprezentuje dvojmiestne vozidlo, dedí od triedy, ktorá reprezentuje štvormiestne vozidlo, a prekonáva metódu umiestni() tak, že v prípade umiestňovania osoby na miesto číslo 1 danú osobu umiestni aj na miesto č. 3, a v prípade umiestňovania osoby na miesto č. 2, danú osobu umiestni aj na miesto č. 4.

Týmto sa dôsledky a predpoklady pôvodnej metódy zoslabujú, zosilňujú alebo nemenia? Je týmto dodržaný Liskovej princíp substitúcie (LSP)? Je v tomto prípade vhodne použité dedenie?

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

Priezvisko:

Meno:

13. (10 b) V hre vystupujú dobrý a zlý čarodejník, ktorí môžu získať a použiť rôzne druhy predmetov, akými sú napríklad čarovný prútik a zlatý prsteň. Ak dobrý čarodejník použije čarovný prútik, vznikne ohňostroj, a ak použije zlatý prsteň, vznikne dúha. Ak zlý čarodejník použije čarovný prútik, vznikne blesk, a ak použije zlatý prsteň, vznikne búrka. Druhy predmetov môžu pribúdať. Nie je vylúčené, že pribudnú aj ďalšie druhy čarodejníkov, ale to je menej očakávané.

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. *Samotné efekty (ohňostroj, dúha atď.) iba naznačte – neimplementujte ich.*

Identifikujte explicitne prvky, ktorými sú modelované a implementované roly aplikovaného návrhového vzoru. 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 2022/23 **A**
doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU
Skúška – 31. máj 2023
30

1 a

2 e

3 d

4 b

5 c

6 b

7 a

8 d

9 FA2F12F12

10 e

11 CxCy AxBxAy AxBxAyBy CxAyBy AxBxAy (akceptovaná je aj odpoveď bez medzier)

12 nemenia sa / zoslabujú sa / nedodržaný / nie

13 Vhodný je vzor Visitor. Druhy čarodejníkov by boli odvodené od spoločnej triedy alebo rozhrania, ktorá by bola v role Elementu. Druhy predmetov by boli odvodené od spoločnej triedy alebo rozhrania, ktorá by bola v role Visitora.