

(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. (2 b) Daná je trieda T v Jave:

```
class A {
    void z() throws Q {
        ...
    }
    void m() throws R {
        try {
            z();
        } catch (Q q) {
            ...
        }
        ...
    }
}
```

Z tohto kódu možno usúdiť, že

- (a) metóda m() môže vyhodíť výnimku typu R a Q
- (b) metóda m() vždy vyhadzuje výnimku typu R
- (c) metóda m() vždy vyhadzuje výnimku typu Q
- (d) metóda m() môže vyhodíť výnimku typu Q
- (e) metóda m() môže vyhodíť výnimku typu R

2. (1 b) Analogickým mechanizmom k šablónam (templates) v jazyku C++ v Jave

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

3. (3 b) Trieda kruh je odvodená od triedy elipsa. Metódy na nastavenie ohnísk a polomerov sú prekonané tak, aby obe ohníská a oba polomery pri zmene hociktorého z nich mali rovnakú hodnotu. Týmto sa predpoklady a dôsledky týchto metód zoslabujú, zosilňujú alebo sa nemenia?

Odpovedzte vo forme

predpoklady ... / dôsledky ...

pričom tri body nahradíte jednou z nasledujúcich možností:

- zoslabujú sa
- zosilňujú sa
- nemenia sa

4. (1 b) V jazyku C# udalosti

- (a) jestvujú ako mechanizmus jazyka a možno ich použiť aj mimo GUI
- (b) nemožno implementovať
- (c) nepredstavujú mechanizmus jazyka
- (d) jestvujú ako mechanizmus jazyka, ale nemožno ich použiť mimo GUI
- (e) sú to isté čo mechanizmus delegátov

5. (2 b) V návrhovom vzore Visitor

- (a) element je odvodený od visitora
- (b) visitor je odvodený od elementu
- (c) element nepriamo rozširuje visitor o nové metódy
- (d) visitor nepriamo rozširuje element o nové metódy
- (e) element a visitor sú nezávislé

6. (2 b) Daný je nasledujúci program v Jave:

```
class A extends Thread {
    C c;
    public A(C c) {
        this.c = c;
    }
    public 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 void run() {
        for (int i = 0; i < 9999; i++)
            c.b();
    }
}
class C {
    private char a = 'a', b = 'b';
    public synchronized void a() {
        if (a != b)
            System.out.println("a");
        a = 'a';
        b = 'a';
    }
    public void b() {
        if (a != b)
            System.out.println("b");
        a = 'b';
        b = 'b';
    }
    public static void main(String[] args) {
        C c = new C();
        new A(c).start();
        new B(c).start();
    }
}
```

Výstupom tohto programu

- (a) bude desaťtisíckrát znak b
- (b) budú desaťtisíckrát znaky a a b v nepravidelnom striedaní
- (c) nebude nič alebo budú znaky a a b v nepravidelnom počte a striedaní
- (d) bude desaťtisíckrát znak a
- (e) nebude nič

7. (1 b) Potenciálnou nevýhodou aspektov v jazyku AspectJ je, že

- (a) nedokážu ovplyvniť používateľské rozhranie
- (b) nedokážu pridať nové prvky do tried
- (c) je možné ich mať iba v malom počte
- (d) nedokážu nahradiť existujúce metódy v triedach
- (e) v ovplyvnenom kóde táto skutočnosť nie je viditeľná

8. (1 b) Je možné len pomocou medzitypových deklarácií (bez použitia RTTI) v jazyku AspectJ ovplyvniť správanie programu?

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

9. (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 objektovo-orientovaného návrhu je to, že

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

10. (3 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
interface I {
    void m();
}
abstract class P implements I {
    public void m() {
        System.out.print("P");
    }
}
class Q extends P {
    public void m() {
        System.out.print("Q");
    }
}
class R extends Q {
    public void m() {
        super.m();
        System.out.print("R");
    }
}
class C {
    public static void exe(I... a) {
        for (I e : a)
            e.m();
    }
    public static void main(String[] args) {
        P x = new R();
        P y = new Q();
        R z = new R();
        I w = (I) new Q();

        exe(x, y, (P)z, (Q)z, w);
    }
}
```

11. (1 b) Mechanizmus friend v jazyku C++

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

(vypláte tlačným písmom)

Priezvisko:

Meno:

12. (10 b) V systéme na správu úloh každá úloha má názov, začiatok realizácie, predpokladané trvanie a opis. Úloha môže nadväzovať na (jednu) inú, rodičovskú úlohu. Okrem iných, na evidované úlohy sú poskytované tieto dva pohľady, ktoré sa automaticky aktualizujú pri zmene údajov úloh:

- zoznam názvov úloh so začiatkom realizácie
- nadväznosť úloh – napr. zoznam dvojíc (*úloha, rodičovská úloha*)

Nakreslite diagram tried s najvýznamnejšími vzťahmi, operáciami a atribútmi, ktoré vyplývajú z uvedeného opisu vnútorného modelu programu (GUI nie je predmetom otázky).

Napište zodpovedajúci kód v Jave vrátane 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. Identifikujte explicitne, čím sú modelované a implementované roly aplikovaného vzora a vysvetlite, čo sa týmto vzorom dosahuje.

Objektovo-orientované programovanie

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

Skúška – 28. máj 2015

B

1 e

2 b

3 nemenia sa / zoslabujú sa

4 a

5 d

6 c

7 e

8 b

9 a

10 QRQQRQRQ

11 d

30