

(vyplňte tlačенým písmom)

Priezvisko:

Meno:

1 b	
2 b	
3 b	

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Skúška trvá 75 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 píšete len písmeno, ktorým je označená odpoveď, ktorú vyberáte. Odpoveď na otázku 13 píšete výlučne na list, na ktorom sa nachádza jej znenie. Poškodený list nebude uznaný.

1. (3 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) použiť zapuzdrenie
- (b) použiť polymorfizmus
- (c) implementovať ovládanie myšou namiesto klávesnicou
- (d) vyčleniť kód jednotlivých prípadov (case) do zodpovedajúcich metód
- (e) ponechať všetko tak, ako je

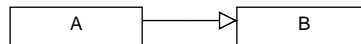
2. (2 b) Okno v rámci Swing sa dá vytvoriť zavolaním

- (a) konštruktora triedy SwingWindow
- (b) metódy newWindow() triedy JFrame
- (c) statickej metódy EventQueue.newWindow()
- (d) statickej metódy SwingUtils.newWindow()
- (e) konštruktora triedy JFrame

3. (1 b) V implementácii vzoru Observer v jazyku AspectJ

- (a) logika vzoru je vyčlenená do aspektu, kým aplikačná logika zostáva v triedach
- (b) aplikačná logika a logika vzoru zostávajú v triedach
- (c) aplikačná logika je vyčlenená do aspektu, kým logika vzoru zostáva v triedach
- (d) aplikačná logika a logika vzoru sú zmiešané v triedach a aspektoch
- (e) aplikačná logika a logika vzoru sú vyčlenené do aspektu

4. (2 b) Daný je nasledujúci diagram v jazyku UML:



Znázornený vzťah sa na úrovni implementácie v Jave prejaví tak, že

- (a) trieda B bude dedič od triedy A
- (b) trieda A bude dedič od triedy B
- (c) trieda A bude obsahovať triedu B
- (d) trieda A bude vplyvať na triedu B
- (e) trieda B bude obsahovať triedu A

5. (2 b) Podľa Liskovej princípu substitúcie

- (a) treba čím viac používať dedenie
- (b) pri použití dedenia treba dbať na kód, ktorým je implementované
- (c) pri použití dedenia treba dbať na klientsky kód
- (d) treba čím menej používať dedenie
- (e) pri použití dedenia treba dbať na reálne alebo matematické objekty

6. (1 b) V jazyku C++ prekonávanie sa použitím špeciálneho kľúčového slova

- (a) môže v prípade, že nie je potrebné, deaktivovať
- (b) musí v prípade potreby aktivovať
- (c) musí v prípade, že nie je potrebné, deaktivovať
- (d) musí v prípade potreby aktivovať, a následne sa môže deaktivovať
- (e) musí v prípade potreby aktivovať, a následne sa musí aj deaktivovať

7. (3 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
abstract class A {
    public static int i;
    public void m() {
        System.out.print("a");
    }
}
class B extends A {
    public void m() {
        System.out.print("b");
        i++;
    }
}
class C extends B {
    public void m() {
        System.out.print("c");
        i--;
    }
}
class M {
    public static void main(String[] args) {
        new B().m();
        new C().m();
        ((B)new C()).m();
        ((A)new B()).m();
        ((A)new C()).m();
        System.out.print(A.i);
    }
}
```

8. (3 b) Daný je nasledujúci kód v Jave:

```
abstract class A {
}
class X extends A {
}
class Y extends X {
}
class M {
    static void m(Class<? extends A> T, A... o) {
        int i = 0;
        for (A e : o) {
            if (T.isInstance(e))
                i++;
        }
        System.out.print(i);
    }
    public static void main(String[] args) {
        m(Y.class, new A[]{new X(), (X)new Y(),
            (A)new Y()});
    }
}
```

Aká hodnota sa vypíše po jeho vykonaní?

9. (2 b) Metóda f() triedy A vyhadzuje výnimku MyException. Daná je trieda B:

```
class B {
    void m() {
        new A().f();
    }
}
```

Metóda m() triedy B

- (a) musí vyhadzovať výnimku typu MyException
- (b) musí deklarovať, že vyhadzuje výnimku typu MyException
- (c) musí deklarovať alebo ošetrovať výnimku typu MyException
- (d) musí ošetrovať výnimku typu MyException
- (e) je korektná

10. (3 b) Daný je nasledujúci program v Jave:

```
class A {
    private int i1 = 1, i2 = 1;
    public void m() {
        if (i1 == 1) {
            i1 = 2;
            i2 = 2;
        }
        else {
            i1 = 1;
            i2 = 1;
        }
    }
}
class B implements Runnable {
    A a;
    public B(A a) {
        this.a = a;
    }
    public void run() {
        for (int i = 1; i < 100000; i++)
            a.m();
    }
}
class M {
    public static void main(String[] args) {
```

```
A a = new A();
new Thread(new B(a)).start();
new Thread(new B(a)).start();
}
}
```

Aby hodnota atribútu i1 bola vždy rovnaká ako hodnota atribútu i2

- (a) metóda m() musí byť synchronizovaná
- (b) metóda main() musí byť synchronizovaná
- (c) metóda run() musí byť synchronizovaná
- (d) metóda m() a metóda run() musia byť synchronizované
- (e) metóda run() a metóda main() musia byť synchronizované

11. (2 b) Daný je nasledujúci kód v Jave:

```
interface A {
    void m(X x);
}
interface X {
    ...
}
class C implements A {
    public void m(X x) {
        x.op(this);
    }
}
```

Pre ktorý návrhový vzor je tento kód charakteristický?

12. (1 b) V jazyku C# sa pri práci s vlastnosťami (properties) prístupové a nastavovacie funkcie (set/get) volajú

- (a) implicitne pri modifikácii, a explicitne pri prístupe
- (b) implicitne alebo explicitne
- (c) explicitne
- (d) implicitne pri prístupe, a explicitne pri modifikácii
- (e) implicitne

(vypláte tlačným písmom)

Priezvisko:

Meno:

13. (10 b) V systéme sa okrem iného uchováva zoznam textových položiek (spôsob implementácie zoznamu zvolte; položky môžu byť obmedzenej veľkosti). Položky je vo všeobecnosti možné filtrovať podľa rôznych kritérií, výsledkom čoho je zoznam položiek, ktoré toto kritérium spĺňajú. Spôsoby filtrovania majú tendenciu pribúdať a to v ľubovoľnej programovej zložitosti.

Nakreslite diagram tried s najvýznamnejšími vzťahmi, operáciami a atribútmi, ktoré vyplývajú z uvedeného opisu vnútorného modelu systému (GUI nie je predmetom otázky). Zahrňte dva špecifické spôsoby filtrovania. Napíšte zodpovedajúci kód v Jave vrátane (vykonštruovaného) príkladu použitia, v ktorom demonštrujete použitie filtrovania. Aplikujte adekvátne mechanizmy objektovo-orientovaného programovania. Ak je to vhodné, aplikujte niektorý z návrhových vzorov a vysvetlite, čo sa ním dosahuje.

Objektovo-orientované programovanie

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

Opravná skúška – 1. júl 2013

B

1 d

2 e

3 a

4 b

5 c

6 b

7 bccbc-1

8 2

9 c

10 a

11 Visitor

12 e

35