

Základy objektovo-orientovaného programovania

A

Ing. Ján Lang, PhD., UIIS FIIT STU

Test - 9. novembra 2017 - riadny termín

Meno a priezvisko:

1	
2	
3	
4	
5	
6	

Test trvá 30 minút. Do tabuľky uveďte písmeno pod ktorým je označená odpoveď, ktorú vyberáte, ak samotná otázka neurčuje iný spôsob odpovedania. V uzavretých otázkach s ponúknutými odpoveďami je vždy správna iba jedna možnosť. Hodnotia sa len odpovede v tabuľke. V prípade opravy jasne vyznačte odpoveď, ktorá platí. Každá správna odpoveď má hodnotu vyznačenú v otázke. Nesprávna odpoveď, alebo nejednoznačné vyznačenie má hodnotu 0 bodov. Postup riešenia sa nehodnotí. Akceptovaný bude len odovzdaný celistvý list.

1. (2b) Daný je nasledujúci kód v Jave:

```
class P {
    void m(N d) {
        System.out.print("P");
    }
}

public class A extends P {
    public void m(N d) {
        super.m(d);
        d.op(this);
        System.out.print("A");
    }
}

public class B extends P {
    public void m(N d) {
        super.m(d);
        d.op(this);
        System.out.print("B");
    }
}

public class N {
    void op(A a) {
        System.out.print("NA");
    }

    void op(B b) {
        System.out.print("NB");
    }
}
```

Do tabuľky k úlohe vpíšte to čo vypíše nasledovný kód:

```
public static void main(String[] args) {
    A o1 = new A();
    o1.m(new N());
    P o2 = new P();
}
```

2. (1b) Z kódu uvedeného v úlohe č. 1 je zrejmé že objekty triedy A disponujú nasledovnými vzťahmi k objektom iných tried:

- (a) Generalizácie
- (b) Generalizácie a asociácie
- (c) Generalizácie a agregácie
- (d) Generalizácie a kompozície
- (e) Agregácie a kompozície
- (f) Agregácie a asociácie
- (g) Kompozície a asociácie
- (h) Kompozície

3. (2b) Daný je nasledujúci kód v Jave:

```
public class A {
    A() {
        System.out.print("A");
    }
}

public class B extends A {
    B(A a) {
        super();
        System.out.print("B");
    }
}

public class C extends B {
    C() {
        super(new A());
        System.out.print("C");
    }
}
```

Do tabuľky k úlohe vpíšte to čo vypíše nasledovný kód:

```
public static void main(String[] args) {
    new C();
    new B(new A());
    new A();
}
```

4. (2b) Daný je kód v Jave

```
class Soup {
    Soup() {}
    private static Soup ps1 = new Soup();
    public static Soup access() {
        return ps1;
    }
    public void f() {
        System.out.println("It tastes good...");
    }
}
```

Aby sme dosiahli výpis textu: It tastes good... použijeme príkaz:

- (a) Soup.f(new Soup());
- (b) Soup.access(f());
- (c) Soup s = new S(); s.f();
- (d) Soup.access().f();
- (e) Soup.f().access();
- (f) Soup s = new Soup(f());
- (g) Žiaden z uvedených

5. (2b) Daný je nasledujúci kód v Jave:

```
package u04;

public class A {
    public final int m() {
        System.out.print("AA");
        return 0;
    }
    public int m(int i) {
        System.out.print("AB");
        return 0;
    }
}

package u05;

import u04.A;

public class B extends A {
    public int m(String s) {
        System.out.print("BA");
        return 0;
    }
    public final int m(int i) {
        System.out.print("BB");
        return 0;
    }
}
```

Metóda `public final int m(int i)` triedy B:

- (a) Neprekonáva rovnomennú metódu triedy A nakoľko je finálna
- (b) Nepreťažuje rovnomennú metódu triedy A
- (c) Nepreťažuje rovnomennú metódu triedy B
- (d) Prekonáva rovnomennú metódu triedy A
- (e) Prekonáva rovnomennú metódu triedy B
- (f) Žiadna z uvedených možností nie je správna

6. (1b) Deklarácia `import static java.lang.Math.*;`

- (a) Naimportuje všetky statické atribúty a metódy triedy `Math`
- (b) Naimportuje všetky statické atribúty triedy `Math`
- (c) Naimportuje všetky statické metódy triedy `Math`
- (d) Sprístupní priestor názvov statických atribútov triedy `Math`
- (e) Sprístupní priestor názvov statických atribútov a metód triedy `Math`
- (f) Sprístupní priestor názvov statických metód triedy `Math`

spolu 10 bodov

Riešenie:

1	PNAA	2b
2	b	1b
3	AABCAABA	2b
4	d	2b
5	d	2b
6	e	1b