

Základy objektovo-orientovaného programovania A

doc. Ing. Ján Lang, PhD., UISI FIIT STU

Test - 7. novembra 2023 - riadny termín

Meno a priezvisko:

1		Test trvá 30 minút. V uzavretých otázkach s ponúknutými odpoveďami je vždy správna iba jedna možnosť. Do tabuľky uveďte písmeno pod ktorým je označená odpoveď, ktorú vyberáte. 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.
2		
3		
4		
5		
6		

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

```
public class Account {
    private String userID = null;

    Account(String s) {
        this.userID = s;
    }

    public final void modifyID(String modifyID)
    {
        this.userID = modifyID;
    }
}

public class specialAccount extends Account {
    specialAccount(String s) {
        super(s);
    }

    public void modifyID(String modifyID,
double amount) {
    }
}

public static void main(String[] args) {
    Account a = new Account("A2016_1011");
}
```

Nový účet:

- vznikne
- nevznikne kvôli tomu, že userID je už inicializovaný na null
- nevznikne, pretože metóda modify() je finálna
- nevznikne, lebo konštrukcia super(s); nemôže pristupovať k finálnej metóde
- nevznikne pretože trieda Account nemá bezparametrický konštruktor
- vznikne po úprave Account a = new Account(new "A2016_1011");
- nevznikne kvôli zlému formátu argumentu pri volaní konštruktora

2. (2b) Daný je nasledujúci kód v Java:

```
public class A {
    protected int i;

    public A(int i) {
        System.out.println("Toto je A");
    }

    public void m() {
        System.out.println("metoda m");
    }

    public void m(int i) {
        System.out.println("iná metoda m");
    }
}

public class B extends A {
    public final void m() {
        super.m();
        super.i++;
        System.out.println("metoda m triedy B");
    }
}
```

Problémom tohto kódu je:

- chýbajúci parametrický konštruktor v triede B
- pokus o inkrementáciu atribútu i
- nesprávne použitie kľúčového slova final
- chýbajúci bezparametrický konštruktor triedy A
- pokus o prekonanie metódy m()
- duplicita v pomenovaní metód triedy A
- žiadna z uvedených možností

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

```
public class M {
    public int m() {
        return 0;
    }

    public int m(int i) {
        return 0;
    }
}

public class N extends M {
    public int m(int i, int j) {
        super.m();
        return 0;
    }
}
```

Ktoré z nasledujúcich tvrdení týkajúce sa kódu vyššie je pravdivé?

- metóda m triedy N preťažuje metódu m triedy M
- metóda m triedy N prekonáva metódu m triedy M
- metóda m triedy N nemôže preťažiť implicitnú metódu m triedy M
- metóda m triedy N nemôže prekonať finálnu metódu m triedy M kvôli volaniu super.m();
- metódu m triedy N je možné v triede M prekonať
- volanie super.m(); nie je korektné
- žiadna z uvedených možností

4. (2b) Do tabuľky k úlohe vpište to čo vypíše nasledovný kód:

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

    void m(){
        System.out.print("N");
    }

    static void n() {
        System.out.print("O");
    }
}

public class N extends M {
    public N() {
        System.out.print("O");
    }

    void m() {
        System.out.print("N");
    }

    static void n() {
        System.out.print("M");
    }
}

public class O extends N {
    public O() {
        System.out.print("M");
    }

    void m(){
        System.out.print("N");
    }

    static void n() {
        System.out.print("O");
    }
}

public class Main {
    public static void main(String[] args)
    {
        new O().m();
        new M();
        N.n();
        new N().m();
        O.n();
        new O();
    }
}
```

5. (1b) Daný je nasledujúci kód v Java:

```
public class Kosik {
    Zelenina z;

    public void pridaj(Zelenina z) {
        this.z = z;
    }
}

public class Zelenina {
}

public class Mrkva extends Zelenina {
}

public class Kalerab extends Zelenina {
}

public class Main {
    public static void main() {
        Kosik k1 = new Kosik();
        k1.pridaj(new Zelenina());
        Kosik k2 = new Kosik();
        Zelenina z = new Zelenina();
        k2.pridaj(z);
    }
}
```

Vzťah košíka k1 a zeleniny pripomína:

- (a) dedenie
- (b) agregáciu
- (c) kompozíciu
- (d) preťaženie
- (e) prekonanie
- (f) zapuzdrenie

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

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

spolu 10 bodov

Riešenie:

1	a	2b
2	d	2b
3	a	2b
4	MOMNMMMONOMOM	2b
5	b	1b
6	f	1b