

Základy objektovo-orientovaného programovania **B**

Ing. Ján Lang, PhD., UISI FIIT STU

Test - 12. novembra 2015 - riadny termín

Meno a priezvisko:

1	
2	
3	
4	
5	
6	
7	
8	
9	

Test trvá 35 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.

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

```
public class A {
    void m(int i) {
    }
    void m(int i, int j) {
    }
}

public class B extends A {
    void n(int i) {
    }
    void m(int i, int j) {
    }
}
```

Ktoré z nasledovného je pravdivé tvrdenie?

- (a) A.m(int i, int j) prekonáva A.m(int i)
- (b) B.m(int i, int j) preťažuje A.m(int i)
- (c) B.m(int i, int j) preťažuje B.n(int i)
- (d) B.n(int i) prekonáva A.m(int i)
- (e) B.n(int i) preťažuje A.m(int i)
- (f) B.m(int i, int j) prekonáva B.n(int i)
- (g) B.m(int i, int j) prekonáva A.m(int i)

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

```
package sk.stuba.fiit.priklad05a;
public class A {
    protected int i = 10;
}

package sk.stuba.fiit.priklad05b;
import sk.stuba.fiit.priklad05a.*;
public class B extends A {
    public void pristup() {
        new A().i++;
        new B().i++;
    }
}
```

Inkrementácia atribútu i v metóde pristup() bude:

- (a) úspešná pri inštancii triedy A aj B
- (b) neúspešná ani pri inštancii triedy A ani triedy B
- (c) úspešná pri inštancii triedy B nie však triedy A
- (d) úspešná pri inštancii triedy A nie však triedy B
- (e) metóda pristup() je neprístupná

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

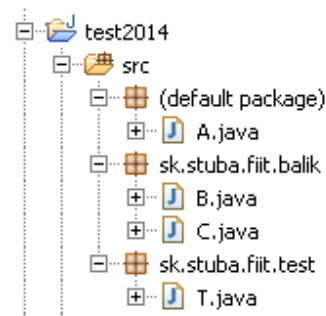
```
public class A {
    public A() {
        System.out.println("A");
    }
}

public class B extends A {
    public B() {
        new A();
        System.out.println("B");
    }
}

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

Aká bude postupnosť vytvárania inštancií tried v uvedenej hierarchii ak v hlavnom programe zadáme príkaz: new C(); Svoju odpoveď zapíšte do riadku pre odpoveď č. 3.

4. (1b) Daný je nasledujúci kód v Jave a rozloženie tried do balíkov v zmysle obr.2:



Obr. 2

```
//...súbor C.java
public class C {
    private int ci=1;
    protected int cj=2;
    public int ck=3;
    int cl=4;
}

//...súbor A.java
import sk.stuba.fiit.balik.C;
public class A extends C {
    public int testDostupnosti() {
        ...todo...
    }
}
```

*Korektný prístup k hodnote atribútu cj triedy C z metódy testDostupnosti() triedy A je:

- (a) System.out.println(cj);
- (b) System.out.println(A().cj);
- (c) System.out.println(C().cj);
- (d) System.out.println(new C().cj);
- (e) System.out.println(new T().cj);
- (f) žiaden z uvedených

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

```
public class A {  
    public double m() {  
        return 0.0;  
    }  
}  
public class B extends A {  
    double m() {  
        return 100;  
    }  
}
```

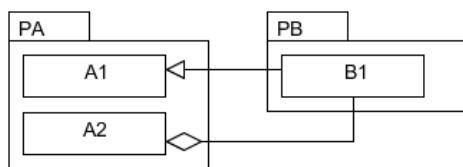
Problém metódy m() triedy B je:

- (a) nesprávna viditeľnosť metódy m() triedy B
- (b) v prekonávaní metódy m() triedy A
- (c) v nepoužitom preťažení metódy m() triedy A
- (d) vo viditeľnosti metódy m() triedy A
- (e) nesprávna syntax metódy m() triedy B
- (f) nesprávna návratová hodnota metódy m() triedy B

6. (2b) Čo z nasledovného nie je charakteristikou vzťahu kompozície?

- (a) Každý celok môže byť považovaný za zoskupenie častí
- (b) Časť môže byť zdieľaná viacerými celkami v konkrétnom čase
- (c) Konkrétna časť obyčajne zaniká zánikom celku
- (d) Časť musí vždy patriť nejakému celku
- (e) Časť niekedy nemôže byť oddelená od pôvodného celku
- (f) Žiadne z uvedeného

7. (2b) Daný je nasledujúci diagram v jazyku UML:



Obr.1

Ktoré z nasledovného môže reprezentovať vzťah medzi triedou A2 a triedou B1 z obr. 1?

- (a) `new B1();`
- (b) `A2 a2 = new B1();`
- (c) `new B1().prístup(A2 a2);`
- (d) `public class A2 extends B1 { ... }`
- (e) `public class B1 extends A1 { ... }`
- (f) `new A2().prístup(B1 b1);`

8. (2b) Daný je nasledujúci kód v Jave:

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

Ktoré z nasledujúcich tvrdení týkajúce sa kódu vyššie je pravdivé?

- (a) metóda m triedy N prekonáva metódu m triedy M
- (b) metóda m triedy N nemôže preťažiť finálnu metódu m triedy M
- (c) metóda m triedy N nemôže preťažiť finálnu metódu m triedy M kvôli volaniu super.m();
- (d) metódu m triedy N je možné v triede M prekonať
- (e) volanie super.m(); nie je korektné
- (f) žiadne z uvedených

9. (2b) Daný je nasledujúci kód v Jave:

```
public class Zelenina {  
    static int pocetKusov = 100;  
}  
public class Hlavna {  
    public static void main(String[] a) {  
        Zelenina paprika = new Zelenina();  
    }  
}
```

Ktoré z nasledujúcich tvrdení týkajúce sa kódu vyššie je pravdivé?

- (a) statický atribút pocetKusov je implicitne prívátny
- (b) objektívne korektný je iba prístup Zelenina.pocetKusov
- (c) inicializácia atribútu je možná len konštruktorom
- (d) na statickú premennú sa dá odkazovať aj pomocou mena objektu triedy Zelenina
- (e) žiadne z uvedených

spolu 15 bodov

Riešenie:

1	b	1b
2	c	2b
3	AABAABC	2b
4	d	1b
5	a	1b
6	b	2b
7	f	2b
8	f	2b
9	d	2b