

Základy objektovo-orientovaného programovania

B

Ing. Ján Lang, PhD., UISI FIIT STU
Skúška - 10. januára 2017

Priezvisko:

Meno:

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Test trvá 60 minút.
V uzavretých otázkach 2-10 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 v otázkach 1-10 nehodnotí. Akceptovaný bude len odovzdaný celistvý list. Riešenie úlohy 11 píšete do prázdneho miesta na liste na ktorom sa nachádza jej znenie. Poškodený list nebude uznaný.

1. (3b) Daný je nasledujúci kód v Jave:

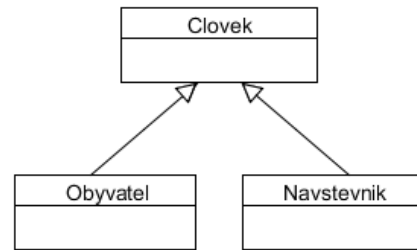
```
public interface I {
    default void m() {
        System.out.println("Calling I.m()");
    }
}

public interface Y {
    default void m() {
        System.out.println("Calling Y.m()");
    }
}

public class A implements I, Y {
    public static void main(String [] args) {
        new A().m();
    }
}
```

Z uvedeného je zjavné, že trieda A bude disponovať aj metódou m. Čo bude obsahovať táto metóda na to aby bolo možné zavolať práve default metódu rozhrania Y? Svoju odpoveď zapíšte do príslušného riadku v tabuľke.

2. (2b) Daná je nasledujúca konštrukcia tried:



Čo môžeme tvrdiť o inštancii o triedy obyvatel' (Obyvatel o = new Obyvatel ();)?

- (a) Objekt o je typu Obyvatel a má k dispozícii všetky metódy z triedy Obyvatel aj Clovek okrem finálnych metód
- (b) Objekt o sa nedá pretypovať na objekt Clovek
- (c) Objekt o sa dá bezpečne pretypovať na objekt Navstevnik
- (d) Kompilátor Java pracuje s objektom o ako s inštanciou triedy Object
- (e) Objekt o je typu Obyvatel a má k dispozícii všetky metódy z triedy Obyvatel aj Clovek vrátane finálnych metód okrem statických metód

3. (3b) Trieda A rozširuje abstraktnú triedu B a implementuje rozhranie I. Metóda m triedy Main disponuje parametrom typu rozhrania. Ktoré z nasledovných tvrdení je správne?

- (a) Nemôžem vytvoriť inštanciu triedy Main a zavolať jej metódu m, ktorej odovzdám ako argument inštanciu triedy A nakoľko tá potrebuje referenciu na rozhranie
- (b) Môžem vytvoriť inštanciu triedy Main a zavolať jej metódu m, ktorej odovzdám ako argument inštanciu triedy A
- (c) Nemôžem vytvoriť inštanciu triedy Main a zavolať jej metódu m, ktorej odovzdám ako argument inštanciu triedy A nakoľko tá rozširuje abstraktnú triedu
- (d) Môžem vytvoriť inštanciu triedy Main a zavolať jej metódu m pokiaľ táto nie je v kolízii s potenciálnou metódou m triedy A
- (e) Nemôžem vytvoriť inštanciu triedy Main a zavolať jej metódu m, ktorej odovzdám ako argument inštanciu triedy B
- (f) Môžem vytvoriť inštanciu triedy Main a zavolať jej metódu m, ktorej odovzdám ako argument inštanciu triedy B

4. (1b) Upcasting na rozdiel od downcasting-u:

- (a) Sa vykonáva explicitne
- (b) Sa nevykonáva
- (c) Sa vykonáva implicitne
- (d) Mení typ objektu
- (e) Mení typ triedy
- (f) Mení typ rozhrania

5. (3b) Daný je nasledujúci kód v Jave:

```
public class Clovek {  
    ...  
}  
  
public class Obyvatel extends Clovek {  
    ...  
}  
  
public static void main(String[] args) {  
    Obyvatel o = new Obyvatel();  
    Clovek c = new Obyvatel();  
}
```

Ktoré z nasledovných tvrdení je v súvislosti s uvedeným fragmentom kódu správne?

- (a) Referenčná premenná c je definovaná ako Obyvatel
- (b) Priradením c = o; budeme na Cloveka pozerat ako na Obyvateľa
- (c) Objekt referencovaný c nemôže byť pretypovaný na typ Obyvatel
- (d) Objekt referencovaný o môže byť do budúcnosti bez problémov pretypovaný na typ Clovek
- (e) Uvedeným pretypovaním sú skryté všetky špecifické metódy triedy Obyvatel

6. (4b) Daný je nasledujúci kód v Jave:

```
public class A {  
    A() { System.out.print("Ax "); }  
    void m() { System.out.print("Am "); }  
    static void f() { System.out.print("Af "); }  
}  
  
public class B extends A {  
    B() { System.out.print("Bx "); }  
    void m() { System.out.print("Bm "); }  
    static void f() { System.out.print("Bf "); }  
}
```

Čo sa vypíše po vykonaní týchto príkazov:

```
A o = new B();  
o.m();  
o.f();  
((B) o).f();  
((A) o).f();
```

- (a) Ax Bx Bm Af Bf Af
- (b) Am Bx Af Bf Bf Ax
- (c) Bx Bf Bf Bx Am Am
- (d) Ax Af Bf Bx Am Af
- (e) Ax Bx Bm Af Bf Bf
- (f) Bx Af Bf Am Ax Bf
- (g) Am Af Bx Bf Bf Af
- (h) Ax Af Bf Bx Am Am
- (i) Bx Am Af Bf Bf Af

7. (3b) Pretypovanie smerom z nadtypu na podtyp:

- (a) Je možné, je nebezpečné, nepredstavuje syntaktickú chybu a dá sa skompilovať
- (b) Je možné, je nebezpečné, predstavuje syntaktickú chybu a nedá sa skompilovať
- (c) Je možné, je nebezpečné, nepredstavuje syntaktickú chybu a nedá sa skompilovať
- (d) Je možné, nie je nebezpečné, nepredstavuje syntaktickú chybu a dá sa skompilovať
- (e) Je možné, nie je nebezpečné, nepredstavuje syntaktickú chybu a nedá sa skompilovať
- (f) Žiadna z uvedených možností

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

```
public interface Skladovatelny {  
    void skladuj(Sklad s);  
}  
  
public abstract class Potrava implements Skladovatelny {  
    void nakrm(Zviera z) {  
        System.out.println("Som potrava a krmim zviera.");  
    }  
}  
  
public class Kost extends Potrava {  
    void nakrm(Zviera z) {  
        System.out.println("Som Kost a krmim zviera.");  
    }  
}  
  
public abstract class Zviera {  
    void zjedz(Potrava p) {  
        System.out.println("Zviera");  
    }  
}  
  
public class Sklad {  
    void pridajDoSkladu(Potrava p) {  
    }  
}
```

Nasledovná metóda

```
public void skladuj(Sklad s) {  
    s.pridajDoSkladu(this);  
}
```

s názvom skladuj:

- (a) Nemôže byť implementovaná v triede Kost
- (b) Musí byť implementovaná v triede Potrava
- (c) Môže byť implementovaná v triede Kost
- (d) Nemôže byť implementovaná v triede Potrava
- (e) Musí byť implementovaná v triede Kost
- (f) Žiadna z uvedených možností

Priezvisko:**Meno:**

9. (1b) V prípade, že sa rovnomenné metódy - zdedená a pridaná v podtype nelíšia v parametroch hovoríme o:

- (a) Agregácii
- (b) Prekonaní
- (c) Preťažení
- (d) Zapuzdrení
- (e) Dedení

10. (2b) Daný je nasledujúci kód v Jave:

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

Ktoré z nasledovných tvrdení je správne?

- (a) konštruktor v triede B nesmie byť implicitný
- (b) konštruktor v triede B musí byť parametrický
- (c) konštruktor v triede B nesmie byť parametrický
- (d) konštruktor v triede B môže byť implicitný
- (e) konštruktor v triede B musí byť a bezparametrický

11. (10b) Na položenú otázku resp. zadanú úlohu možno nazerať ako na udalosť. Otázky môžu byť otvorené (doplňovacie), alebo uzavreté - dichotomické (áno/nie), polytomické (s výberom odpovede) a pod. Klienti nášho testovacieho centra sa môžu prihlasovať na rôzne certifikačné (testovacie) programy. Absolvovaním testu je klientovi vystavený certifikát s definovanou dobou platnosti. Výška poplatkov za test a certifikát bude variabilná pre rôzne kategórie. Súčasťou centra sú aj kontrahovaní autori testov. Systém okrem iného umožní pridávanie otázok do testov, testovanie, výpočet poplatkov, vystavovanie certifikátov a pod. Napíšte zodpovedajúci kód v Jave. Mapujte reálne entity virtuálneho sveta a aplikujte adekvátne mechanizmy objektovo-orientovaného programovania. Špeciálne uplatnite polymorfizmus. Napokon nakreslite diagram identifikovaných tried s uvedením vzťahov medzi triedami. Uplatnené mechanizmy OOP v kóde viditeľne vyznačte.

Spolu 35 bodov

Riešenie:

	bodov	
1	3	<code>Y.super.m();</code>
2	2	d
3	3	b
4	1	c
5	3	e
6	4	a
7	3	a
8	3	e
9	1	b
10	2	d