Základy objektovo-orientovaného programovania **A**
Ing. Ján Lang, PhD., UISI FIIT STU
Test - 13. novembra 2013

Priezvisko:
Meno:

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	

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

```
public class Prva {
	public Prva() {
		System.out.print("P");
	}
}
public class Druha extends Prva{
	public Druha() {
		System.out.print("D");
	}
}
public class Tretia extends Prva {
	public Tretia() {
		System.out.print("T");
	}
}
```

Čo sa vypíše po vykonaní príkazov:

```
new Druha();
new Tretia();
new Prva();
```

(a) DPPTP
(b) PDPTP
(c) DPTPP
(d) PTPDP
(e) DPPPT

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

```
public class A {
	static int pocet = 0;
	int p = 1;

	public A(int pocet) {
		this.pocet++;
		this.p = pocet;
		pocet = 3;
	}
}
```

Čo sa vypíše po vykonaní príkazov:

```
new A(2);
System.out.println(A.pocet);
```

(a) 0
(b) 1
(c) 2
(d) 3
(e) Kompilačná chyba

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

```
class Clovek {
String meno, priezvisko;

Clovek(String meno, String priezvisko) {
 this.meno = meno;
 this.priezvisko = priezvisko;
}

public String toString() {
 return meno + " " + priezvisko;
}

void vypis() {
 System.out.println(priezvisko + " " + meno);
 }
}

class Student extends Clovek {
 int rocnik;

Student(String meno, String priezvisko, int
rocnik) {
 super(meno, priezvisko);
 this.rocnik = rocnik;
}

public String toString() {
 return priezvisko + " " + meno;
}

void vypis() {
 System.out.println(meno + " " + priezvisko);
 }
}
```

Ktorý z príkazov zabezpečí výpis na konzolu v tvare:

```
Adam Prvy
```

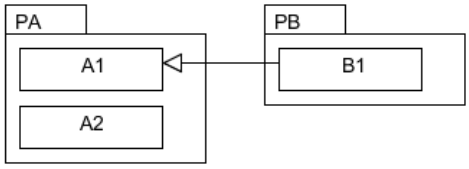
(a)
```
System.out.println(new Student("Adam",
"Prvy", 1));
```
(b)
```
new Clovek("Adam", "Prvy").vypis();
```
(c)
```
System.out.println(new Student("Adam",
"Prvy", 1).vypis());
```
(d)
```
new Student("Adam", "Prvy",
1).super.vypis();
```
(e)
```
System.out.println(new Clovek("Adam",
"Prvy"));
```

4. (1b) Inštancia triedy
(a) má svoj stav a správanie ale nemá identitu
(b) nemá stav, ale má správanie a identitu
(c) má svoj stav, správanie a identitu
(d) má identitu, stav ale ešte nemá správanie
(e) nemá identitu ani stav ale má správanie

5. (1b) Stav objektu triedy
(a) je definovaný hodnotami jeho atribútov
(b) je definovaný bezparametrickým konštruktorom
(c) je definovaný inštanciou agregovanej metódy
(d) je definovaný modifikátormi prístupu
(e) je nedefinovaný

6. (1b) Medzi triedou A a triedou B je vzťah agregácie. Tento vzťah:
(a) reprezentuje dedenie vlastností nadtypu
(b) znamená dedenie vlastností podtypu
(c) umožňuje skrývanie implementácie objektu
(d) predstavuje princíp znovupoužitia programového kódu skladaním
(e) poskytuje možnosť meniť vlastnosti oboch tried

7. (1b) Daný je nasledujúci diagram v jazyku UML:



Znázornený vzťah na úrovni implementácie v jazyku Java znamená, že
(a) atribúty triedy A1 sú v rámci triedy B1 viditeľné s uvedením modifikátora `public` a `protected`
(b) atribúty triedy B1 sú v rámci triedy A1 viditeľné s uvedením modifikátora `public` a `protected`
(c) atribúty triedy A1 sú v rámci triedy B1 viditeľné s uvedením modifikátora `private` a `protected`
(d) atribúty triedy B1 sú v rámci triedy A1 viditeľné s uvedením modifikátora `private` a `protected`
(e) atribúty triedy A1 niesú v rámci triedy B1 viditeľné s uvedením modifikátora `public` a `protected`
(f) atribúty triedy B1 sú v rámci triedy A1 viditeľné iba s uvedením modifikátora `public` a `protected`

8. (1b) Rozhranie poskytované objektom
(a) reprezentuje stav inštancie
(b) je súborom jeho atribútov
(c) popisuje množinu inštancií triedy, ktoré majú rovnaké vlastnosti a správanie
(d) vymedzuje požiadavky, ktoré naň môžu byť kladené
(e) popisuje množinu inštancií triedy, ktoré nemajú rovnaké vlastnosti a správanie

9. (1b) Trieda
(a) popisuje množinu objektov, ktoré majú rovnaké atribúty a správanie. Stav jednotlivých objektov však nemôže byť odlišný
(b) popisuje množinu objektov, ktoré majú rovnaké vlastnosti a správanie. Stav jednotlivých objektov však nemôže byť odlišný
(c) popisuje množinu objektov, ktoré nemajú rovnaké vlastnosti a správanie. Stav jednotlivých objektov však môže byť odlišný
(d) popisuje množinu objektov, ktoré nemajú rovnaké vlastnosti a správanie. Stav jednotlivých objektov však nemôže byť odlišný
(e) popisuje množinu objektov, ktoré majú rovnaké vlastnosti a správanie. Stav jednotlivých objektov však môže byť odlišný

10. (1b) Deklarácia nestatickej metódy rovnakej signatúry v Podtype
(a) preťažuje (overloads) pôvodnú metódu Nadtypu
(b) umožňuje jej dedenie tým, že je nestatická
(c) prekonáva (overrides) pôvodnú metódu Nadtypu
(d) neumožňuje jej dedenie práve preto, že je nestatická
(e) preťažuje (overloads) pôvodnú metódu Podtypu

11. (1b) Deklarácia import static java.lang.Math.*;
(a) Sprístupní priestor názvov statických atribútov a metód triedy Math
(b) Sprístupní priestor názvov statických atribútov triedy Math
(c) Sprístupní priestor názvov statických metód triedy Math
(d) Naimportuje všetky statické atribúty a metódy triedy Math
(e) Naimportuje všetky statické atribúty triedy Math
(f) Naimportuje všetky statické metódy triedy Math

12. (1b) Dedenie nám v odvodenej triede neumožňuje:
(a) ponechať a využívať všetko z rodičovskej triedy
(b) doplniť, čo v rodičovskej triede nebolo a v odvodenej chýba
(c) zmeniť to, čo z rodičovskej triedy nevyhovuje
(d) preťažovať metódy rodičovskej triedy
(e) meniť štruktúru rodičovskej triedy

13. (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)	`B.n(int i)`	prekonáva	`A.m(int i)`
(b)	`A.m(int i, int j)`	prekonáva	`A.m(int i)`
(c)	`B.m(int i, int j)`	preťažuje	`B.n(int i)`
(d)	`B.m(int i, int j)`	preťažuje	`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)`

Spolu 15 bodov
Riešenie:

1	b
2	b
3	e
4	c
5	a
6	d
7	a
8	d
9	e
10	c
11	a
12	e
13	d