

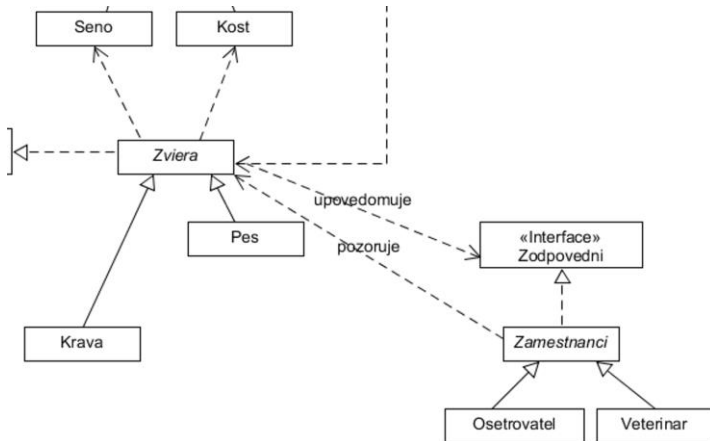
Priezvisko:

Meno:

1	
2	
3	
4	
5	
6	
7	
8	
9	

Test trvá 60 minút.
 V uzavretých otázkach 1-9 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 1-9 nehodnotí. Akceptovaný bude len odovzdaný celistvý list. Riešenie úlohy 10 píše do prázdneho miesta na liste na ktorom sa nachádza jej znenie. Poškodený list nebude uznaný.

1. (4b) Vzťah *upovedomuje* je realizovaný:



- Volaním metódy predpisovanej rozhraním *Zodpovedni* referenciou typu *Seno* alebo *Kost*
- Volaním metódy predpisovanej rozhraním *Zodpovedni* referenciou typu *Zodpovedni*
- Volaním metódy predpisovanej rozhraním *Zodpovedni* referenciou typu *Krava* alebo *Pes*
- Volaním metódy predpisovanej rozhraním *Zodpovedni* referenciou výhradne typu *Osetrovatel* alebo *Veterinar*
- Volaním metódy predpisovanej rozhraním *Zodpovedni* referenciou výhradne typu *Zamestnanci*
- Volaním metódy predpisovanej abstraktnou triedou *Zviera* referenciou typu *Zodpovedni*
- Volaním metódy predpisovanej abstraktnou triedou *Zodpovedni* referenciou typu *Zviera*
- Žiadne z uvedených

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

```

public interface I {
    public void m();
}

public class A implements I {
    public void m() {
    }
}

I i = new A(); //Upcasting
    
```

Ktoré z tvrdení vo vzťahu k vykonaniu kódu je pravdivé:

- Upcasting je mylne interpretovaný, namiesto rozhrania ma byť použitá trieda resp. abstraktná trieda
- Referencia typu interface nikdy nemôže byť inicializovaná inštanciou triedy
- Referencia typu interface môže byť inicializovaná inštanciou triedy iba v prípade abstraktnej triedy
- Trieda A v tomto prípade musí byť abstraktná
- Žiadne z uvedených

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

```

class Student {
    private String meno;
    private int rocnik;

    public Student() {
        this(1); //1 - predpokladá sa prvý
        //ročník
    }

    public Student(int rocnik) {
        this.rocnik = rocnik;
    }
}
    
```

Prvotná inicializácia atribútu *rocnik* objektu triedy *Student* prebehla:

- explicitnou hodnotou pri vytvorení objektu
- implicitným konštruktorom volaným priamo
- explicitným konštruktorom volaným priamo
- explicitným konštruktorom volaným v inom konštruktore tej istej triedy
- formálnym parametrom *rocnik*
- atribútom objektu
- priradením *this.rocnik = rocnik;*

4. (1b) Downcasting predstavuje:

- Implicitnú zmenu podtypu referencie na objekt z typu nadtriedy na typ podtriedy
- Explicitnú zmenu typu referencie na objekt z typu nadtriedy na typ podtriedy
- Implicitnú zmenu podtypu referencie na objekt z typu nadtriedy na typ podtriedy
- Explicitnú zmenu podtypu referencie na objekt z typu nadtriedy na typ podtriedy
- Žiadna z uvedených možností

5. (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");
    }
}
public class Stvrta extends Druha {
    public Stvrta() {
        System.out.print("S");
    }
}
public class Osma extends Stvrta {
    public Osma() {
        System.out.print("O");
    }
}
```

Do tabuľky k úlohe vpište to čo vypíše nasledovný kód:

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

6. (3b) Daný je nasledujúci kód v Jave:

```
public class A {
    void fa(A a) { }
}
public class B extends A {
    void fb(B b) { }
}
public static void main(String[] args) {
    A a = new A();
    A b = new B();
    B c = new B();
}
```

Volanie metódy fb prostredníctvom referencie b:

- (a) Je možné príkazom ((B) b).fb(c);
- (b) Je možné príkazom (B) b.fb(b);
- (c) Je možné príkazom b.fb(b);
- (d) Je možné príkazom b.fb(c);
- (e) Nie je možné

7. (4b) Daný je nasledujúci kód v Jave:

```
class Utvar {
    int farba = 12;
    void nakresli(int farba) {
    }
}
class Kruh extends Utvar {
    void nakresli(int f) {
        System.out.println("Nakresleny " + f);
    }
    void vypln(int f) {
        System.out.println("Vymalovany " + f);
    }
}
Utvar k = new Kruh();
k.vypln(21);
```

Vyberte z nasledovných možností to čo spôsobí kód uvedený vyššie

- (a) vypíše na konzolu: Nakresleny 12
- (b) vypíše na konzolu: Nakresleny 21
- (c) vypíše na konzolu: Vymalovany 12
- (d) vypíše na konzolu: Vymalovany 21
- (e) žiadne z uvedeného

8. (2b) Daný je kód v Jave:

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

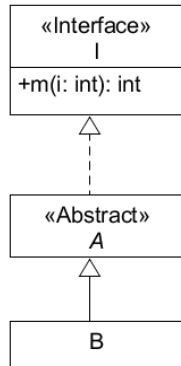
Volanie metódy f() triedy A z rovnomennej metódy triedy B je:

- (a) možné príkazom B.f();
- (b) možné príkazom super.f();
- (c) možné príkazom this.f();
- (d) možné príkazom f();
- (e) žiadne z uvedených

Priezvisko:

Meno:

9. (3b) Daný je diagram v jazyku UML:



Ktoré z tvrdení vo vzťahu k uvedenému diagramu je pravdivé:

- (a) metódu m trieda B nemusí implementovať
- (b) metódu m trieda B nemôže implementovať
- (c) metódu m trieda B musí implementovať iba v prípade, použitia kľúčového slova super
- (d) metódu m trieda B musí implementovať iba v prípade, použitia kľúčového slova final
- (e) metódu m trieda B musí implementovať iba v prípade, použitia kľúčového slova this
- (f) metódu m trieda A musí implementovať iba v prípade, použitia kľúčového slova super
- (g) metódu m trieda A nemôže implementovať
- (h) metódu m trieda A nemusí implementovať
- (i) žiadne z uvedeného

10. (10b) V našom simulátore krajiny máme možnosť počítať a vyberať poplatky za využívanie množiny prvkov dopravnej infraštruktúry (spoplatnené úseky - úseky ciest, mosty, tunely a pod.) vybranými kategóriami dopravných prostriedkov (osobný automobil, autobus, traktor a pod). Výška poplatkov bude variabilná pre rôzne kategórie dopravných prostriedkov a času využitia spoplatnených úsekov (slabé dni, silné dni, sviatkov a pod.). Poplatky bude definovať sadzobník. Systém umožní identifikáciu dopravného prostriedku, výpočet a výber poplatkov, bez nutnosti jeho zastavenia. 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:

1	b	4
2	e	3
3	a	3
4	b	1
5	PPDSOPTPDPDS	2
6	a	3
7	e	4
8	e	2
9	h	3