Kreatívna časť, ukážka 1

Firma ÚINF nutne potrebuje nový portál, keďže na tú starú sa už nedá pozerať. Webový portál pozostáva zo sekcií, pričom každá sekcia obsahuje niekoľko článkov. Článok spadá do jedinej sekcie (napr. *Informácie pre študentov*), ale môže byť označený niekoľkými značkami (napr. *informatika*, *študenti*, *seminár*), podľa ktorých sa dajú články ďalej kategorizovať.

Každý článok obsahuje diskusiu, teda zoznam diskusných príspevkov, ktoré môžu mať stromovitú štruktúru. (Príspevok môže reagovať na iný príspevok).

V portále majú existovať autori článkov, diskutéri a jediný správca. Autor článku vie upravovať svoj článok, diskutér vie len prispievať a správca má neobmedzené možnosti. Samozrejme, roly sa môžu prekrývať – nie je vylúčené, že napr. diskutér má zároveň zverejnený článok.

Navrhnite pre tento informačný systém triedy, uveďte a slovne popíšte ich inštančné premenné a metódy. Zvážte a odôvodnite použitie dedičnosti (ak si myslíte, že dedičnosť je potrebná, slovne to zdôvodnite; ak si myslíte, že potrebná nie je, zdôvodnite to tiež).

Implementujte metódu (t. j. napíšte kód v programovacom jazyku), ktorá zistí zoznam všetkých príspevkov daného používateľa vo všetkých diskusiách. Uveďte aj triedu a relevantné inštančné premenné (v prípade, že ich metóda používa). Inak povedané, napíšte fragment triedy a úplnú implementáciu uvedenej metódy.

Naimplementujte tester (t. j. napíšte kód v programovacom jazyku) s metódou main(), v ktorej ukážete príklady použitia vami navrhnutého informačného systému a tried (predpokladajte, že všetky metódy všetkých tried už máte implementované). V testeri ukážte použitie minimálne 5 metód používaných v triedach informačného systému.

Kreatívna časť, ukážka 2

Národná rada nemenovanej republiky sa rozhodla zaviesť informačný systém. Tento parlament pozostáva zo 150 poslancov, ktoré prináležia niektorej zo strán, ale, samozrejme, existujú v ňom aj nezávislí poslanci. V parlamente prebiehajú hlasovania na ľubovoľnú tému, kde je žiaduce evidovať nielen to, koľko poslancov bolo za, proti, či zdržalo sa, ale aj to, ktorý konkrétny poslanec zaujal aký postoj. Národná rada chce mať navyše prehľad, aké hlasovania prebehli za posledný deň, mesiac a rok, a štatistiku o ich priebehoch. Politická strana chce mať zase prehľad o tom, ako hlasovali ich jednotliví poslanci v hlasovaniach (za posledný deň, mesiac, rok).

Navrhnite pre tento informačný systém triedy, uveďte a slovne popíšte ich inštančné premenné a metódy. Zvážte a odôvodnite použitie dedičnosti (ak si myslíte, že dedičnosť je potrebná, slovne to zdôvodnite; ak si myslíte, že potrebná nie je, zdôvodnite to tiež).

Implementujte metódu (t. j. napíšte kód v programovacom jazyku), ktorá pre daného poslanca a dané hlasovanie zistí zoznam všetkých jeho spolustraníkov, ktorí hlasovali rovnako ako on.

Naimplementujte tester (t. j. napíšte kód v programovacom jazyku) s metódou main(), v ktorej ukážete príklady použitia vami navrhnutého informačného systému a tried (predpokladajte, že všetky

metódy všetkých tried už máte implementované). V testeri ukážte použitie minimálne 5 metód používaných v triedach informačného systému.

Opravovacia časť, ukážka 1

Dve hodiny pred odovzdaním projektu má nemenovaný študent projekt, v ktorom su doleuvedené triedy. Smutne sa pozerá do Eclipsu, ktorý mu podčiarkol značné množstvo kódu, ktoré je chybné. Popri tom je však v kóde aj niekoľko logických či návrhových chýb.

Projekt má totiž sčítavať obsahy trojuholníkov, čo vykonáva nesprávne (vracia mu to záporné čísla). Vžite sa do úlohy úbohého cvičiaceho, zistite všetky chyby a opravte ich. Ak vidíte chyby v návrhu, zdôvodnite ich slovne a navrhnite riešenie. Zároveň opravte logické chyby, aby to nemenovanému študentovi pracovalo tak, ako má.

```
public interface MaObsah {
	int dajObsah() {
		return -1;
	}
}
```

```
public class Usecka {
	int x, y, popis;
	public Usecka(int x, int y) {
		this.x = x;
		this.y = y;
	}
	public String toString() {
		return popis + " [" + x + ", " + y + "]";
	}
}
```

```
public class Trojuholnik extends Usecka implements MaObsah {
	private int z;
	//urobi konstruktor s danymi bodmi
	public Trojuholnik(int A, int B, int C) {
		super();
		this.x = A;
		this.y = B;
		this.z = C;
	}
	// TODO: duri mi poradil Heronov vzorec
	// S = sqrt(s(s-a)(s-b)(s-c)), s = (a + b + c)
	// netusim ako ho pouzit
	public int getObsah() {
		return x * y * z;
	}
}
```

[kód pokračuje na ďalšej strane]

```
public class Tester {
	public static int spocitajObsahy(List<Usecka> trojuholniky) {
		int q = 0;
		for (int i = 0; i < trojuholniky.size(); i++) {
			Trojuholnik t = trojuholniky.get(i);
			q = q + t.dajObsah();
		}
		return obsahy;
	}

	public static void main(String[] args) {
	Trojuholnik[] = new ArrayList<Usecka>();
	Trojuholnik t1 = new Trojuholnik(2, 4, 5);
	trojuholniky.add(t1);
	Trojuholnik t2 = new Trojuholnik(4, 5, 6);
	trojuholniky.add(t2);
	obsahy = spocitajObsahy(trojuholniky);
	// tu mi to vypisuje -2 netusim preco
	System.out.println(obsahy);
	}
}
```

Opravovacia časť, ukážka 1

```
public class Matica {
	private int[][] matica = new ArrayList<ArrayList<int>>();

	public matica(int hura, int hura2) {
		matica = new int[hura][hura2];
	}
	Public getPocetRiadkov(matica Matica) {
		Return matica.length;
	}
	Public getPocetStlpcou(matica Matica) {
		Return matica[0].length;
	}
	public int pocetStlpcov() {
		return matica[0].length;
	}

	Public int nastavPrvok(int chobotina, chobotina2, Prvok prvok)
	{
		 Matica[chobotina][chobotina2] = prvok;
	}
}
```

[kód pokračuje na ďalšej strane]

```
public class vektor extends matica {
	public vektor() {
		//vytvori prazdny vektor
		super();
	}
	Public vektor(int rozmer) {
		matica.super(10,0);
	}
	Public getPocetStlpcou(matica Matica) {
		return 0;
	}
	Public getPocetRiadkov() {
		Return matica[0].length
	}

	// odmocnina (x_1*x_1 ... x_n * x_n)
	Public dajNormu() {
		Int norma = 0;
		For(i = 0; i < matica[0].length) {
			Norma = norma + matica.prvky.get(i);
		}
		Return Math.sqrt(norma);
	}
}
```

```
public class jednotkovamatica extends matica {
	Public int nastavPrvok(int chobotina, chobotina2, Prvok prvok} {
		if(chobotina == chobotina 2) {
			if(prvok != 1) {
			throw new IllegalArgumentException(
				“Na diagonále musí byť jednotka!“);
}}}}
```

```
import java.util.List;
public static void main(String[] args) {
	// chcem zratat sucet matic v poli
	List<jednotkovamatica> matice = new List<jednotkovamatica>;
	matice.get(0).nastavPrvok(2, 3, 2, 3);
	// dalej nestiham, neviem pokracovat
}
```