**Міністерство освіти і науки, молоді та спорту України**

**Національний технічний університет України**

**«Київський політехнічний інститут»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**Лабораторна робота №5**

**Тема: «Відношення між класами в мові програмування Java.»**

**Виконав: Студент групи ІО-33**

**Заколенко Р.К.**

**Перевірив: Викладач**

**Невдащенко М.В.**

**Київ 2014**

1. **Завдання**

1.Модифікувати лабораторну роботу №3 наступним чином: для літер, слів, речень, розділових знаків та тексту створити окремі класи. Слово повинно складатися з масиву літер, речення з масиву слів та розділових знаків, текст з масиву речень. Замінити послідовність табуляцій та пробілів одним пробілом.

2.Створити клас, який складається з виконавчого методу, що виконує описану дію з лабораторної роботи №3 (з кожного речення заданого тексту видалити підрядок найбільшої довжини, що починається та закінчується заданими літерами), але в якості типів використовує створені класи.

1. **Опис програми**

Програма містить шість класів:

1. Клас Letter;
2. Клас Punctuation;
3. Клас Word;
4. Клас Sentence;
5. Клас Text;
6. Клас Lab5(із виконавчим методом).

У виконавчому методі ми створюємо об’єкт класу Text, за допомогою конструктора Text(String s), де параметр «s»-наш заданий текст, потім створюємо два об’єкти класу Letter, за допомогою конструктора Letter(char a),параметр «a» задає поле «letter» для кожного об’єкту. Далі за допомогою метода removeOfString(Text t, Letter a, Letter b) виконуємо дію з лабораторно роботи №3. Відредагований текст буде виведено на екран.

1. **Код програми**

**package** ua.kpi.io33.zakolenko.lab5;

/\*\*

\* This class describe data about object Letter.

\*

\* **@author** Roman Zakolenko

\* **@version** 1,6 22 Apr 2014

\*/

**public** **class** Letter {

/\*\*

\* Field "letter" which houses the meaning of the letter.

\*/

**private** **char** letter;

/\*\*

\* **@param** a

\* this variable is assigned to the field "letter".

\*/

**public** Letter(**char** a) {

letter = a;

}

/\*\*

\* **@return** value of fields "letter".

\*/

**public** **char** getLetter() {

**return** letter;

}

/\*\*

\* **@param** letter

\* is assigned to the field "letter".

\*/

**public** **void** setLetter(**char** letter) {

**this**.letter = letter;

}

}

**package** ua.kpi.io33.zakolenko.lab5;

/\*\*

\* This class describe data about object Punctuation.

\*

\* **@author** Roman Zakolenko

\* **@version** 1,6 22 Apr 2014

\*/

**public** **class** Punctuation {

/\*\*

\* Field "symbol" which houses the meaning of the punctuation.

\*/

**private** **char** symbol;

/\*\*

\* Field "position" which houses the meaning of the position of punctuation

\* of sentence.

\*/

**private** **int** position;

/\*\*

\* **@param** a

\* this variable is assigned to the field "symbol".

\* **@param** p

\* this variable is assigned to the field "position".

\*/

**public** Punctuation(**char** a, **int** p) {

setSymbol(a);

setPosition(p);

}

/\*\*

\* **@return** value of field "symbol".

\*/

**public** **char** getSymbol() {

**return** symbol;

}

/\*\*

\* **@param** symbol

\* is assigned to the field "symbol".

\*/

**public** **void** setSymbol(**char** symbol) {

**this**.symbol = symbol;

}

/\*\*

\* **@return** field "position".

\*/

**public** **int** getPosition() {

**return** position;

}

/\*\*

\* **@param** position

\* is assigned to the field "position".

\*/

**public** **void** setPosition(**int** position) {

**this**.position = position;

}

}

**package** ua.kpi.io33.zakolenko.lab5;

/\*\*

\* This class describe data about object Word.

\*

\* **@author** Roman Zakolenko

\* **@version** 1,6 22 Apr 2014

\*/

**public** **class** Word {

/\*\*

\* Field "word" which houses the meaning of the word.

\*/

**private** String word;

/\*\*

\* An array of objects of class Letter, of which the word.

\*/

**private** Letter[] letters;

/\*\*

\* **@param** a

\* this variable is assigned to the field "word".

\*/

**public** Word(String a) {

setWord(a);

**this**.letters = **new** Letter[a.length()];

**for** (**int** i = 0; i < a.length(); i++) {

letters[i] = **new** Letter(a.charAt(i));

}

}

/\*\*

\* **@return** value of field "word".

\*/

**public** String getWord() {

**return** **new** String(word);

}

/\*\*

\* **@param** word

\* is assigned to the field "word".

\*/

**public** **void** setWord(String word) {

**this**.word = **new** String(word);

}

/\*\*

\* **@return** field "letters".

\*/

**public** Letter[] getLetters() {

**return** letters.clone();

}

/\*\*

\* **@param** letter

\* is assigned to the field "letters".

\*/

**public** **void** setLetters(Letter[] letter) {

**this**.letters = letter.clone();

}

}

**package** ua.kpi.io33.zakolenko.lab5;

**import** java.util.ArrayList;

/\*\*

\* This class describe data about object Sentence.

\*

\* **@author** Roman Zakolenko

\* **@version** 1,6 22 Apr 2014

\*/

**public** **class** Sentence {

/\*\*

\* Field "sent" which houses the meaning of the sentence.

\*/

**private** String sent;

/\*\*

\* An array of objects of class Word, of which the sentence.

\*/

**private** Word[] words;

/\*\*

\* The list of objects of class Punctuation that contains punctuation of the

\* sentence.

\*/

**private** ArrayList<Punctuation> p;

/\*\*

\* **@param** a

\* this variable is assigned to the field "sent".

\* **@param** s

\* this variable is assigned to the field "p".

\*/

**public** Sentence(String a, **char** s) {

setSent(a + s);

ArrayList<Punctuation> p1 = **new** ArrayList<Punctuation>();

**for** (**int** i = 0; i < a.length(); i++) {

**if** (a.charAt(i) == ',' || a.charAt(i) == ';' || a.charAt(i) == ':'

|| a.charAt(i) == '-') {

p1.add(**new** Punctuation(a.charAt(i), i));

}

}

p1.add(**new** Punctuation(s, a.length()));

p = p1;

String sent1[] = a.split("[,-:;]\\s|\\s[,-:;]|[,-:;]|\\s");

**int** k = 0;

**for** (**int** i = 0; i < sent1.length; i++) {

**if** (sent1[i].isEmpty()) {

k++;

}

}

**this**.words = **new** Word[sent1.length - k];

**for** (**int** i = 0, j = 0; i < sent1.length; i++) {

**if** (!sent1[i].isEmpty()) {

words[j] = **new** Word(sent1[i]);

j++;

}

}

}

/\*\*

\* **@return** value of field "sent".

\*/

**public** String getSent() {

**return** **new** String(sent);

}

/\*\*

\* **@param** sent

\* is assigned to the field "sent".

\*/

**public** **void** setSent(String sent) {

**this**.sent = **new** String(sent);

}

/\*\*

\* **@return** field "words".

\*/

**public** Word[] getWords() {

**return** words.clone();

}

/\*\*

\* **@param** word is assigned to the field "words".

\*/

**public** **void** setWord(Word[] word) {

**this**.words = word.clone();

}

/\*\*

\* **@return** field "p".

\*/

**public** ArrayList<Punctuation> getP() {

**return** **new** ArrayList<Punctuation>(p);

}

/\*\*

\* **@param** p is assigned to the field "p";

\*/

**public** **void** setP(ArrayList<Punctuation> p) {

**this**.p = **new** ArrayList<Punctuation>(p);

}

}

**package** ua.kpi.io33.zakolenko.lab5;

/\*\*

\* This class describe data about object Text.

\*

\* **@author** Roman Zakolenko

\* **@version** 1,6 22 Apr 2014

\*/

**public** **class** Text {

/\*\*

\* Field "txt" which houses the meaning of the text.

\*/

**private** String txt;

/\*\*

\* An array of objects of class Sentence, of which the text.

\*/

**private** Sentence[] sentence;

/\*\*

\* **@param** s

\* this variable is assigned to the field "txt".

\*/

**public** Text(String s) {

String s1 = s.replaceAll("\\s+", " ");

setTxt(s1);

;

String sent1[] = s1.split("\\.|!|\\?");

**this**.sentence = **new** Sentence[sent1.length];

**int** z = 0;

**for** (**int** i = 0; i < sent1.length; i++) {

z = z + sent1[i].length();

sentence[i] = **new** Sentence(sent1[i], s1.charAt(z + i));

}

}

/\*\*

\* **@return** value of field "txt".

\*/

**public** String getTxt() {

**return** **new** String(txt);

}

/\*\*

\* **@param** txt

\* is assigned to the field "txt".

\*/

**public** **void** setTxt(String txt) {

**this**.txt = **new** String(txt);

}

/\*\*

\* **@return** field "sentence".

\*/

**public** Sentence[] getSentence() {

**return** sentence.clone();

}

/\*\*

\* **@param** sentence

\* is assigned to the field "sentence".

\*/

**public** **void** setSentence(Sentence[] sentence) {

**this**.sentence = sentence.clone();

}

}

**package** ua.kpi.io33.zakolenko.lab5;

/\*\*

\* Is used to create objects of the class Text and Letter, as well as to remove

\* from the specified substring of this string.

\*

\* **@author** Roman Zakolenko

\* **@version** 1,7 07 Apr 2014

\* **@see** ua.kpi.io33.zakolenko.lab5.Text ua.kpi.io33.zakolenko.lab5.Sentence

\* ua.kpi.io33.zakolenko.lab5.Word ua.kpi.io33.zakolenko.lab5.Letter

\* ua.kpi.io33.zakolenko.lab5.Punctuation

\*/

**public** **class** Lab5 {

/\*\*

\* Method that removes from the specified string specified substring.

\*

\* **@param** t

\* associated with an object of class Text.

\* **@param** a

\* indicates the beginning of the substring.

\* **@param** b

\* indicates the end of the substring.

\*/

**public** **static** **void** removeOfString(Text t, Letter a, Letter b) {

**boolean** k = **true**;

StringBuffer s = **new** StringBuffer();

**for** (**int** i = 0; i < t.getSentence().length; i++) {

**int** a1 = t.getSentence()[i].getSent().indexOf(a.getLetter());

**int** b1 = t.getSentence()[i].getSent().lastIndexOf(b.getLetter());

**if** (a1 != -1 && b1 != -1) {

s.append(t.getSentence()[i].getSent().substring(0, a1)

+ t.getSentence()[i].getSent().substring(b1 + 1));

k = **false**;

} **else** {

s.append(t.getSentence()[i].getSent());

}

}

**if** (k) {

System.*out*.println("В данной строке не найдено заданых символов.");

} **else** {

Text t1 = **new** Text(s.toString());

System.*out*.println("Вы ввели текст:" + "\n" + t.getTxt());

System.*out*.println("Отредактированный текст:");

System.*out*.println(t1.getTxt());

}

}

**public** **static** **void** main(String[] args) {

String s = "Толкин подготовил несколько карт Средиземья и отдельных его областей, "

+ "где проходят события его произведений! Не все они были опубликованы "

+ "при жизни.";

Text b = **new** Text(s);

Letter a = **new** Letter('н');

Letter k = **new** Letter('й');

*removeOfString*(b, a, k);

}

}

**4.Висновок**

В результаті виконання лабораторної роботи я детальніше ознайомився із класами в мові програмування Java. Було створено 5 класів(Punctuation, Letter, Word, Sentence, Text), об’єкти яких були використані для виконання дії з лаб.роботи №3. Також було створено клас із виконавчим методом, у якому відбувається редагування заданого тектсу.

Для заміни послідовності табуляцій та пробілів одним пробілів був використан метод replaceAll(“\\s+”,” ”). Для створення масиву об’єктів класу Punctuation, який є полем класу Sentence було використано структуру ArrayList: при знаходженні у реченні розділового знаку створюється новий об’єкт класу Punctuation, який додається у список.

Був створений метод removeOfString(Text t, Letter a, Letter b), який виводить на екран відредагований текст, якщо у реченнях були знайдені задані літери, інакше на екран буде виведено «В данном тексте не найдено заданных символов».

Для захисту інформації поля створених класів мають директиву private, тому були створені «гетери» і «сетери», у яких перед повертанням або встановленням значення створюється новий об’єкт, який не посилається на дане поле.