МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №5

з дисципліни «Об’єктно-орієнтоване програмування»

на тему: *«Відношення між класами в мові програмування Java»*

Виконав:

студент гр.ІО-31

Долинний О.В.

Перевірив:

Невдащенко М.В.

Київ 2014

**Лабораторна робота №5**

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

**Мета:** Ознайомлення з відношеннями між класами в мові програмування Java. Здобуття навичок у використанні відношень між класів в мові програмування Java.

1. ***Варіант завдання***

Номер залікової книжки  *3110*

Тип текстових змінних: *C3 = 3110 mod 3 = 0* String

Дія з рядком: *C17 = 3110 mod 17 = 16*

В кожному слові заданого тексту, видалити всі попередні входження останньої літери цього слова.

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, але в якості типів використовує створені класи. Необхідно обробити всі виключні ситуації, що можуть виникнути під час виконання програмного коду. Всі змінні повинні бути описані та значення їх задані у виконавчому методі. Код повинен відповідати стандартам JCC та бути детально задокументований.

1. ***Опис алгоритму програми***

Клас *Main* містить виконавчий метод, що в кожному слові заданого рядка з видаляє попередні входження останньої літери цього слова.

У класі *Main* створюється екземпляр класу *Text,* метод *setText* розбиває текст на речення, метод *setSentence* розбиває речення на слова та розділові знаки, *setWord* – розбиває слова на літери. Текст – це масив речень, речення – масиви слів та розділових знаків, слово – масив літер.

Також у класі *Main* послідовність табуляцій та пробілів замінюється одним пробілом.

Метод *Deleting\_Letter()* у класі *Word* видаляє попередні входження літери, на яку закінчується слово.

У класах *Text, Sentence, Word, Punctuation* був перевизначений *toString().*

1. ***Програмний код***

Class Main

/\*

\* @(#)Main.java 1.0 23/04/14

\*

\* Copyright (c) 2014 Alex Dolinniy

\*/

**package** lab\_5;

/\*\* Class Main was created to do 5 lab on Programming.

\* The task consists in editing of text:

\* deleting all previous entries the latest letter in word

\* of these word in all words of these text.

\*

\* **@version** 1.0 23 April 2014

\* **@author** Alex Dolinniy

\* **@since** 1.0

\*/

**public** **class** Main {

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

String inputText = **new** String(" Besarss, leeike hoyneyy. " +

"Teehe snun: iss gshigningg briyghtyylyy!");

/\*Changing text according to the task\*/

Text text = **new** Text();

text.setText(inputText);

String result = text.toString();

System.*out*.println("\nAnswer: \n" + result);

}

}

Class Text

/\*

\* @(#)Text.java 1.0 23/04/14

\*

\* Copyright (c) 2014 Alex Dolinniy

\*/

**package** lab\_5;

/\*\* Class Text creates object and divides it

\* into array of sentences.

\* Class has setter, constructor

\* and overridden method toString.

\*

\* **@version** 1.0 23 April 2014

\* **@author** Alex Dolinniy

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** text

\* array of sentences

\* **@param** countSentence

\* quantity of sentences

\*/

**private** Sentence[] text;

**private** **int** countSentence=0;

/\*\* Constructor of Text

\*/

**public** Text() {

text = **new** Sentence[30];

**int** countSentence = 0;

}

/\*\* Setter

\*/

**public** **void** setText(String str) {

/\* Deleting tabulations and several spaces\*/

String res = **new** String();

**for** (**int** i = 0; i<str.length()-1; i++) {

**char** c = str.charAt(i);

**if** ((c=='\n')||(c=='\t'))

c=' ';

**if** ((c!=' ') || (((c==' ')) && ((str.charAt(i+1)!=' ')

&& (str.charAt(i+1)!='\n') && (str.charAt(i+1)!='\t'))))

res+=c;

}

res+=str.charAt(str.length()-1);

**int** pos=0;

**for** (**int** i = pos; i < res.length(); i++) {

**char** c = res.charAt(i);

**if**((c == '.')||(c == '?')||(c == '!')) {

String sub = res.substring(pos+1, i+1);

Sentence buf = **new** Sentence();

String temp = **new** String(sub);

buf.setSentense(temp);

text[countSentence++] = buf;

pos = i + 1;

}

}

}

@Override

**public** String toString() {

String result = "";

**for** (**int** i = 0; i < countSentence; i++)

result = result + text[i].toString();

**return** result;

}

}

Class Sentence

/\*

\* @(#)Sentence.java 1.0 23/04/14

\*

\* Copyright (c) 2014 Alex Dolinniy

\*/

**package** lab\_5;

/\*\* Class Sentence creates object and divides it

\* into array of words and punctuations.

\* Class has getter, setter, constructor

\* and overridden method toString.

\*

\* **@version** 1.0 23 April 2014

\* **@author** Alex Dolinniy

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** sentence

\* array of words

\* **@param** punct

\* array of punctuation symbols

\* **@param** countWord

\* quantity of words

\*/

**private** Word[] sentence;

**private** Punctuation[] punct;

**private** **int** countWord=0;

/\*\*Constructor of Sentence\*/

**public** Sentence() {

sentence = **new** Word[30];

punct = **new** Punctuation[30];

**int** countWord = 0;

}

/\*\*Setter\*/

**public** **void** setSentense(String str) {

/\* position of start of copy symbol to word\*/

**int** pos = 0;

/\* Dividing sentence into words + punctuations\*/

**for** (**int** i = pos; i < str.length(); i++) {

/\* Symbol in buffer\*/

String sub = **new** String();

**if**((str.charAt(i) == ' ')||(i==str.length()-1)) {

/\* Situation when word ends on punctuation\*/

**if** ((str.charAt(i-1) == '.')||(str.charAt(i-1) == '?')

||(str.charAt(i-1) == '!')||(str.charAt(i-1) == ',')

||(str.charAt(i-1) == ':')||(str.charAt(i-1) == ';')) {

punct[countWord] = **new** Punctuation(str.charAt(i-1));

sub = str.substring(pos, i-1);

}

/\* Situation when word doesn't end on punctuation\*/

**else**

sub = str.substring(pos, i);

/\* Situation with last letter of string\*/

**if** (i==str.length()-1) {

punct[countWord] = **new** Punctuation(

str.charAt(str.length()-1));

sub = str.substring(pos, i);

}

Word buf = **new** Word();

buf.setWord(sub);

sentence[countWord++] = buf;

pos = i+1;

}

}

}

@Override

**public** String toString() {

String result = "";

**for** (**int** i = 0; i < countWord; i++)

**if** ( punct[i]!=**null** )

result = result + sentence[i].getWord() + punct[i]+ " ";

**else**

result = result + sentence[i].getWord()+ " ";

**return** result;

}

}

Class Word

/\*

\* @(#)Word.java 1.0 23/04/14

\*

\* Copyright (c) 2014 Alex Dolinniy

\*/

**package** lab\_5;

/\*\* Class Text creates object and divides it

\* into array of letters.

\* Class has getter, setter, constructor,

\* method of deleting all previous entries the latest letter

\* in word and overridden method toString. \*

\*

\* **@version** 1.0 23 April 2014

\* **@author** Alex Dolinniy

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** word

\* array of letters

\* **@param** countLetter

\* quantity of letters

\*/

**private** Letter[] word;

**private** **int** countLetter;

/\*\*Constructor of Word\*/

**public** Word() {

word = **new** Letter[30];

countLetter = 0;

}

/\*\* Getter\*/

**public** String getWord() { // метод, що повертає слово,конвертоване у строку

String str;

str = Deleting\_Letter();

**return** str.toString();

}

/\*\* Setter\*/

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

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

word[countLetter++] = **new** Letter(str.charAt(i));

}

/\*\* Method deletes previous entries the latest letter in word\*/

**public** String Deleting\_Letter(){

/\*input string\*/

String str = **new** String();

str = toString();

/\*output string\*/

String result = **new** String();

/\* Converting word to array of letters \*/

**char** arr[] = **new** **char** [str.length()] ;

str.getChars(0, str.length(), arr, 0) ;

/\* Deleting letter\*/

**for** (**int** i=0; i<=(arr.length-1); i++) {

**if** ((arr[i]!=arr[arr.length-1])&&(arr[i]!=(arr[arr.length-1]-32)))

result+=arr[i];

**if** (i==arr.length-1)

result+=arr[i];

}

**return** result;

}

@Override

**public** String toString() {

String result = **new** String();

**for** (**int** i = 0; i < countLetter; i++)

result+=word[i].getLetter();

**return** result;

}

}

Class Letter

/\*

\* @(#)Letter.java 1.0 23/04/14

\*

\* Copyright (c) 2014 Alex Dolinniy

\*/

**package** lab\_5;

/\*\* Class Text creates object of letter.

\* Class has getter, setter, constructor.

\*

\* **@version** 1.0 23 April 2014

\* **@author** Alex Dolinniy

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** letter

\* symbol of letter

\*/

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

/\*\*Constructor of Letter\*/

**public** Letter(**char** inLetter) {

setLetter(inLetter);

}

/\*\* Getter\*/

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

**return** letter;

}

/\*\* Setter\*/

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

letter = inLetter;

}

}

Class Punctuation

/\*

\* @(#)Punctuation.java 1.0 23/04/14

\*

\* Copyright (c) 2014 Alex Dolinniy

\*/

**package** lab\_5;

/\*\* Class Text creates object of punctuation symbol.

\* Class has getter, setter, constructor.

\*

\* **@version** 1.0 23 April 2014

\* **@author** Alex Dolinniy

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** punctuation

\* symbol of punctuation

\*/

**private** **char** punctuation;

/\*\*Constructor of symbol of punctuation \*/

**public** Punctuation(**char** inPunctuation) {

setPunctuation(inPunctuation);

}

/\*\* Getter\*/

**public** **char** getPunctuation() {

**return** punctuation;

}

/\*\* Setter\*/

**public** **void** setPunctuation(**char** inPunctuation) {

punctuation = inPunctuation ;

}

@Override

**public** String toString() {

**return** "" + punctuation ;

}

}

***Висновок***

Під час лабораторної роботи я закріпив знання із використання класів, відношеннями між ними в мові програмування Java, розвинув навички коментування коду.

Код був оформлений і задокументований відповідно до стандартів JCC.