Національний Технічний Університет України

«Київський Політехнічний Інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №5

*Відношення між класами в мові програмування Java*

|  |  |
| --- | --- |
| Прийняв  Ст.в. Невдащенко М. В.  «\_\_»\_\_\_\_\_\_\_\_ 2014 р. | Виконала Студентка 1-ого курсу ФІОТ  групи ІО-32  Шапран К.О. |

1. ***Варіант завдання***

Номер залікової книжки  *3229*

Тип текстових змінних: *C3 = 3229 mod 3 = 1* StringBuffer

Дія з рядком: *C17 = 3229 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.

1. 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.
2. ***Опис алгоритму програми***

Клас *Main* містить виконавчий метод, що в кожному слові заданого рядка 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 *Text*  послідовність пробілів замінюється одним пробілом.

Оброблено виключну ситуацію, коли речення закінчується на «…».

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

1. ***Програмний код***

Class Main

/\*

\* @(#) Main.java 1.0 22/04/14

\*

\* Copyright (c) 2014 Karina Shapran

\*/

**package** out.production.Lab5;

/\*\* Class Main realize the task of Lab5.

\* 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 22 April 2014

\* **@author** Karina Shapran

\* **@since** 1.0

\*/

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

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

StringBuffer str= **new** StringBuffer("Leieefe - leieeeke drggivigngg aa beiceyceele... " +

" ooTo ppkppepep ryrrorurr ebaleancee, uuyuou mtutstt kpeeppp mgogvging!");

System.*out*.println(str);

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

text.setText(str);

System.*out*.println();

text.removeChars();

}

}

Class Text

/\*

\* @(#)Text.java 1.0 22/04/14

\*

\* Copyright (c) 2014 Karina Shapran

\*/

**package** out.production.Lab5;

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

\* into array of sentences.

\* Class has setter, constructor

\* and method, which delete extra spaces.

\*

\* **@version** 1.0 22 April 2014

\* **@author** Karina Shapran

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** sentences

\* array of sentences

\* **@param** numberSentence

\* quantity of sentences

\* **@param** pos

\* position of element

\*/

**private** Sentence[] sentences;

**private** **int** numberSentence = 0;

**int** pos;

/\*\* Constructor of Text\*/

**public** Text() {

sentences = **new** Sentence[100];

}

/\*\* Setter\*/

**public** **void** setText(StringBuffer t) {

StringBuffer s1 = deleteTab(t);

**for** (**int** i = pos; i < s1.length(); i++) {

**char** c = s1.charAt(i);

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

**if** (i + 3 < s1.length() && s1.charAt(i + 1) == '.'

&& s1.charAt(i + 2) == '.') {

String sub = s1.substring(pos, i + 3);

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

sentences[numberSentence] = **new** Sentence();

sentences[numberSentence].setSentence(temp);

numberSentence++;

i = i + 3;

pos = i;

}

**else** {

String sub = s1.substring(pos, i+1);

sub = sub.trim();

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

sentences[numberSentence] = **new** Sentence();

sentences[numberSentence].setSentence(temp);

numberSentence++;

pos = i + 1;

}

}

}

}

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

**public** **void** removeChars (){

**for** (**int** i = 0; i < numberSentence; i++) {

sentences[i].removeChar();

System.*out*.print(" ");

}

}

/\*\* Delete extra spaces\*/

**public** StringBuffer deleteTab(StringBuffer str) {

StringBuffer b = str;

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

**if** (b.charAt(i) == ' ' && b.charAt(i + 1) == ' ') {

b.replace(i, i + 1, "");

i--;

}

**return** b;

}

}

Class Sentence

/\*

\* @(#)Sentence.java 1.0 22/04/14

\*

\* Copyright (c) 2014 Karina Shapran

\*/

**package** out.production.Lab5;

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

\* into array of words and punctuations.

\* Class has setters and constructor.

\*

\* **@version** 1.0 22 April 2014

\* **@author** Karina Shapran

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** words

\* array of words

\* **@param** punctuations

\* array of punctuations

\* **@param** numberWord

\* quantity of words

\* **@param** numberPunct

\* quantity of punctuations

\*/

**private** Word[] words;

**private** Punctuation[] punctuations;

**private** **int** numberWord = 0;

**private** **int** numberPunct=0;

/\*\*Constructor of Sentence\*/

**public** Sentence(){

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

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

}

/\*\*Setter for Sentence\*/

**public** **void** setSentence(StringBuffer so){

String s = so.toString();

String [] strings = s.split(" ");

**char** g;

**char**[] c;

**for** (String word : strings) {

g = word.charAt(word.length()-1);

**if** ((g=='?')||(g=='!')||(g=='.')||(g==',')||(g=='-')||(g==':')){

**if** (word.length()>3&&( word.charAt(word.length()-2)=='.'&&word.charAt(word.length()-3)=='.')){

setWord(word.substring(0,word.length()-3));

setPunct(**new** StringBuffer (s.substring(s.length()-3,s.length())));

}

**else**{

c = **new** **char** [1];

c[0]=g;

setWord(word.substring(0, word.length()-1).trim());

setPunct(**new** StringBuffer(**new** String(c)));

}

}

**else**{

setWord(word.trim());

setPunct(**null**);

System.*out*.print(" ");

}

}

}

/\*\*Setter for Punctuation\*/

**public** **void** setPunct (StringBuffer s){

punctuations[numberPunct] = **new** Punctuation(s);

numberPunct++;

}

/\*\*Setter for Word\*/

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

StringBuffer sl = **new** StringBuffer(str);

words [numberWord] = **new** Word();

words [numberWord].setWord(sl);

numberWord++;

}

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

**public** **void** removeChar (){

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

**if** (punctuations[i]!=**null** & words[i]!=**null** ){

**if** (i!=0)

System.*out*.print(" ");

words[i].removeChar();

System.*out*.print(punctuations[i].getPunct());

}

}

}

**public** String toString(){

String result = "";

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

**if**(i<numberWord-1)

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

**else**

result += words[i].getWord();

**return** result;

}

}

Class Word

/\*

\* @(#)Word.java 1.0 22/04/14

\*

\* Copyright (c) 2014 Karina Shapran

\*/

**package** out.production.Lab5;

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

\* into array of letters.

\* Class has setter and

\* method of deleting all previous entries the latest letter

\* in word.

\*

\* **@version** 1.0 22 April 2014

\* **@author** Karin Shapran

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** letters

\* array of letters

\* **@param** buf

\* string of word

\*/

**private** Letter[] letters;

**private** String buf;

**public** String getWord(){

StringBuffer str = **new** StringBuffer();

**return** str.toString();

}

/\*\* Setter\*/

**public** **void** setWord(StringBuffer sl){

String s = **new** String(sl.toString());

**if** (!s.equals("")){

String word = **new** String(sl);

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

buf = word;

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

letters[i] = **new** Letter(sl.substring(i,i+1));

}

}

}

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

**public** **void** removeChar(){

**if** (buf!=**null**){

**char** c = buf.charAt(buf.length()-1);

StringBuilder str = **new** StringBuilder();

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

**if** (buf.charAt(i)!=c){

str.append(buf.charAt(i));

}

}

str.append(c);

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

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

letters[i]=**new** Letter(str.substring(i,i+1));

}

}

}

**public** String toString(){

String str = " ";

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

str += letters[i].toString();

}

**return** str;

}

}

Class Letter

/\*

\* @(#)Letter.java 1.0 22/04/14

\*

\* Copyright (c) 2014 Karina Shapran

\*/

**package** out.production.Lab5;

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

\* Class has getter, setter, constructor.

\*

\* **@version** 1.0 22 April 2014

\* **@author** Karina Shapran

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** letter

\* symbol of letter

\*/

**private** String letter;

/\*\* Constructor of class Letter\*/

**public** Letter(String ltr){

letter = ltr;

System.*out*.print(letter);

}

/\*\* Getter\*/

**public** String getLetter() {

**return** letter;

}

/\*\* Setter\*/

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

**this**.letter = letter;

}

**public** String toString(){

**return** letter;

}

}

Class Punctuation

/\*

\* @(#)Punctuation.java 1.0 22/04/14

\*

\* Copyright (c) 2014 Karina Shapran

\*/

**package** out.production.Lab5;

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

\*

\* **@version** 1.0 22 April 2014

\* **@author** Karina Shapran

\* **@since** 1.0

\*/

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

/\*\*

\* **@param** punct

\* symbol of punctuation

\*/

**private** StringBuffer punct=**null**;

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

**public** Punctuation(StringBuffer b){

**if** (b!=**null**){

**this**.setPunct(b);

System.*out*.print(b + " ");

}

}

**public** StringBuffer getPunct() {

**if** (punct==**null**){

**return** **new** StringBuffer("");

}**else**

**return** punct;

}

**public** **void** setPunct(StringBuffer punct) {

**this**.punct = punct;

}

}

1. ***Висновок***

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

Код був оформлений і задокументований відповідно до стандартів JCC.