Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №4

Виконав:

студент групи ІО-22

Коломієць О.

Перевірив:

Невдащенко М. В.

Номер залікової книжки:

2209

Київ 2013р.

**Тема**: Класи в мові програмування Java.

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

1) Визначення формули :

2201 mod 3 = 1 => С3 = StringBuffer

2201 mod 17 = 16 => C1 : В кожному слові заданого тексту, видалити всі попередні входження останньої літери цього слова.

2) Опис програми

У класі Main створюється екземпляр класу Text метод setText розбиває текст на речення, метод setSentence розбиває речення на слова, setWord – розбиває слова на літери. Текст – це масив речень, речення – масив слів, слово – масив літер.

Метод isAlpha() перевіряє чи символ літерою, метод Deleting\_Letter() видаляє попередні входження літери, на яку закінчується слово.

3) Код програми

Файл Main

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

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

StringBuffer text = **new** StringBuffer(" Strgigng is combnniation of rcrhar.... OOP is parmmadigm of gprograming?");

text = **new** StringBuffer(text.toString().replaceAll("\n|\r\n" , " "));

text = **new** StringBuffer(text.toString().toLowerCase());

Text txt = **new** Text();

txt.setText(text);

String str = txt.toString();

System.*out*.println("\n Answer: " + str);

}

}

Файл Text

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

**private** Sentence[] text = **new** Sentence[20];//масив речень

**char**[] punct = **new** **char**[10];

**int** pos;

**int** countSentence = 0;

**public** Text(){

pos = 0;

}

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

**return** text;

}

//збереження речення в масив слів

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

**int** index = 0;

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

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

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

String sub = str.substring(pos, i) + str.charAt(i);

punct[index++] = c;

//System.out.println("\n sub: " + sub);

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

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

buf.setSentense(temp);

text[countSentence++] = buf;

pos = i + 1;

}

}

}

// перетворення об'єкту текст у строку

**public** String toString(){

String result = "";

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

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

**return** result;

}

}

Файл Sentence

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

**private** Word[] sentence = **new** Word[15];// масив слів

**int** countWord;

**public** **int** getSize()

{

**return** countWord;

}

//запис речення в масив слів

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

**int** pos = 0;

countWord = 0;

String sub;

**int** N = str.length();

**for** (**int** j = 0; j < N; j++) {

**if** (!*isAlpha*(str.charAt(j))) {

sub = str.substring(pos, j);

pos = j + 1;

**if** (!sub.isEmpty()) {

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

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

buf.setWord(temp);

sentence[countWord++] = buf;

}

}

}

sub = str.substring(pos, N);

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

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

buf.setWord(temp);

sentence[countWord++] = buf;

/\*for (int j = 0; j < countWord; j++){

if(sentence[j] != null)

System.out.println(" \n:" + sentence[j].getWord());

}\*/

}

//перевірка на "літеру "

**public** **static** **boolean** isAlpha(**char** c) {

**if** ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'))

**return** **true**;

**else**

**return** **false**;

}

// перетворення об'єкту речення у строку

**public** String toString(){

String result = "";

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

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

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

**else**

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

**return** result;

}

}

Файл Word

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

**public** Word(){

word = **new** Letter[20];

countLetter = 0;

}

Letter[] word; // масив літер

**int** countLetter; // лічильник кількості літер

**char** temp;

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

StringBuffer str;

str = Deleting\_Letter();

**return** str.toString();

}

// записує слово у масив літер

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

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

**if** (*isAlpha*( str.charAt(i)) ){

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

}

}

}

// функція для визначення літери

**public** **static** **boolean** isAlpha(**char** c) {

**if** ((c >= 'A' && c <= 'Z') || (c >= 'a' && c <= 'z'))

**return** **true**;

**else**

**return** **false**;

}

// метод, що конвертує переворює слово у строку

**public** StringBuffer To\_String(){

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

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

str.append(word[i].getLetter());

}

**return** str;

}

// видалення попереднього входження літери, на яку закінчується кожне слово

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

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

str = To\_String();

**for** (**int** i = 0; i < str.length()-2; i++)

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

str.deleteCharAt(i);

i--;

}

**return** str;

}

}

Файл Letter

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

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

**public** Letter(**char** s){

letter = s; //літера

}

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

**return** letter;

}

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

**this**.letter = letter;

}

}

Файл Punctuation

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

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

**private** **int** index;

**public** Punctuation(**char** s, **int** i){

punctuation = s; // знак пунктуації

index = i; // індекс

}

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

**return** punctuation;

}

**public** **void** setPbnctuation(**char** punctuation) {

**this**.punctuation = punctuation;

}

}

4) Висновок

Під час лабораторної роботи я закріпив знання із використання класів, відношеннями між ними в мові програмування Java, розвинув навички коментування коду, використовуючи такі принципи агрегація та інкапсуляція.