**Утилітарні класи. Обробка масивів і рядків**

**Мета роботи**: Розробка власних утилітарних класів. Набуття навичок вирішення прикладних задач з використанням масивів і рядків.

**Вимоги**

1. Розробити та продемонструвати консольну програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 15 зменшеного на одиницю номера студента в журналі групи.
2. При вирішенні прикладних задач використовувати латинку.
3. Продемонструвати використання об'єктів класу StringBuilder або StringBuffer.
4. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. Helper Class) та для обробки даних використовувати відповідні статичні методи.
5. Забороняється використовувати засоби обробки регулярних виразів: класи пакету java.util.regex (Pattern, Matcher та ін.), а також відповідні методи класу String (matches, replace, replaceFirst, replaceAll, split).

**Розробник**: Рябов Олексій Володимирович КІТ119а №18.

**Загальне завдання**:

Ввести декілька рядків. Розбити на дві групи: рядки, довжина яких менша за середню; рядки, довжина яких не менше середньої. Вивести рядки та їх довжину по групах.

**Опис програми**

**Засоби ООП**: клас, метод классу.

**Структура класів:** один публічний клас Main, один утилітарний клас Helper.

**Важливі фрагменти програми:**

**import** java.util.Scanner;

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

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

Scanner scan= **new** Scanner(System.***in***);

System.***out***.println("Input line: ");

Helper.*start*(**new** StringBuilder(scan.nextLine()));

scan.close();

}

}

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class Helper {

public static void start(StringBuilder ln) {

if(ln == null) {

throw new NullPointerException("line is null");

}

clottingSpace(ln);

ArrayList<StringBuilder> line = findArray(ln);

Task(line);

}

static private ArrayList<StringBuilder> findArray(StringBuilder ln){

ArrayList<StringBuilder> line = new ArrayList<StringBuilder>();

ArrayList<Integer> IndexSpace = new ArrayList<Integer>();

int length = ln.length();

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

if(ln.charAt(i)==' ')

IndexSpace.add(i);

}

if(IndexSpace.size() == 0) {

line.add(new StringBuilder(ln));

}

else

{

line.add(new StringBuilder(ln.substring(0, IndexSpace.get(0))));

IndexSpace.add(length);

for(int i=1;i<IndexSpace.size();i++) {

if(ln.substring(IndexSpace.get(i-1)+1, IndexSpace.get(i))!="")

line.add(new StringBuilder(ln.substring(IndexSpace.get(i-1)+1, IndexSpace.get(i))));

}

}

return line;

}

static private StringBuilder clottingSpace(StringBuilder line) {

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

{

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

line.deleteCharAt(i+1);

i--;

}

}

if(line.charAt(line.length()-1)==' ')

line.deleteCharAt(line.length()-1);

if(line.charAt(0)==' ')

line.deleteCharAt(0);

System.out.println(line);

return line;

}

static private void Task (ArrayList<StringBuilder> line)

{

int average\_size = 0;

for (StringBuilder stringBuilder : line) {

average\_size += stringBuilder.length();

}

average\_size /= line.size();

System.out.println(" Words with more than average letters ");

for (StringBuilder stringBuilder : line) {

if(average\_size < stringBuilder.length())

{

System.out.println("Word - " + stringBuilder.toString() + " count letter " + stringBuilder.length());

}

}

System.out.println(" Words with less than average letters ");

for (StringBuilder stringBuilder : line) {

if(average\_size > stringBuilder.length())

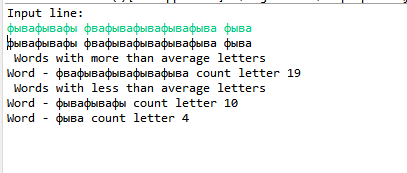
{

System.out.println("Word - " + stringBuilder.toString() + " count letter " + stringBuilder.length());

}

}

**Результати роботи**

****

**Висновки**

Оволодів навичками розробки утилітарних класів. Навчився розробляти методі роботи з масивами та рядками.