Звіт

Автор: Лунгол О., КІТ101.8б

Дата: 24.10.2019

Лабораторна робота №4

ІНТЕРАКТИВНІ КОНСОЛЬНІ ПРОГРАМИ ДЛЯ ПЛАТФОРМИ JAVA SE

**Мета.**

Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою java.

**Вимоги:**

1. Використовуючи програму рішення завдання [лабораторної роботи №3](https://oop-khpi.github.io/#task_03), відповідно до [прикладної задачі](https://oop-khpi.github.io/#task_03_app) забезпечити обробку команд користувача у вигляді текстового [меню](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%BD%D1%8E_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)#.D0.9C.D0.B5.D0.BD.D1.8E_.D0.B2_.D0.B8.D0.BD.D1.82.D0.B5.D1.80.D1.84.D0.B5.D0.B9.D1.81.D0.B5_.D0.BA.D0.BE.D0.BC.D0.B0.D0.BD.D0.B4.D0.BD.D0.BE.D0.B9_.D1.81.D1.82.D1.80.D0.BE.D0.BA.D0.B8):
   * введення даних;
   * перегляд даних;
   * виконання обчислень;
   * відображення результату;
   * завершення програми і т.д.
2. Забезпечити обробку параметрів [командного рядка](https://uk.wikipedia.org/wiki/%D0%86%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D1%80%D1%8F%D0%B4%D0%BA%D0%B0) для визначення режиму роботи програми:
   * параметр “-h” чи “-help”: відображається інформація про автора програми, призначення (індивідуальне завдання), детальний опис режимів роботи (пунктів меню та параметрів командного рядка);
   * параметр “-d” чи “-debug”: в процесі роботи програми відображаються додаткові дані, що полегшують налагодження та перевірку працездатності програми: діагностичні повідомлення, проміжні значення змінних, значення тимчасових змінних та ін.

ЗАВДАННЯ ДО РОБОТИ

Ввести текст. Визначити та вивести, яких літер (голосних чи приголосних) більше в кожному реченні тексту. Результат вивести у вигляді таблиці.

ОПИС ПРОГРАМИ

**Опис змінних**

Scanner scan = **new** Scanner(System.***in***);//змінна для сканування введеної інформації

String str1 = **new** String();// створення нового порожнього рядка

**int** index1//індекс знаку «.»

**int** v = 0//змінна для голосних літер

**int** sign = 0;//змінна для знаків

**Ієрархія та структура класів**

**class** Lab04 – точка входу в програму

**class** ArgsCMD – клас для командного рядка

**class** Function - клас з основними функціями

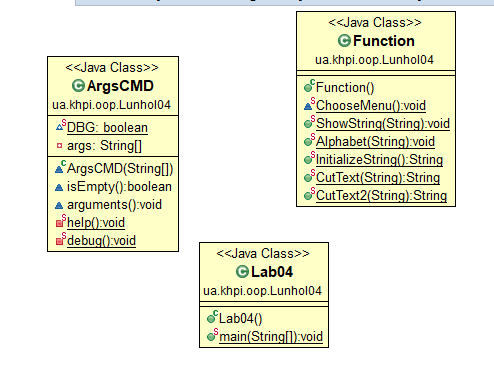


Рисунок 1 – Класи програми

ТЕКСТ ПРОГРАМИ

Lab04.java

**package** ua.khpi.oop.Lunhol04;

**import** java.util.Arrays;

**import** java.util.Scanner;

**import** **static** ua.khpi.oop.Lunhol04.Function.\*;

**public** **class** Lab04 {

/\*\*

\* **@author** ALEXANDRA

\* **@version** 4.0

\* Here start point of the program

\* Function in which menu actions are performed

\* until the user presses exit program does not end

\*

\* \*/

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

ArgsCMD Arguments = **new** ArgsCMD(args);

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

Arguments.arguments();

}

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

**boolean** loop = **true**;

**int** choose;

String str1 = **null**;

String str2 = **null**;

**while** (loop) {

*ChooseMenu*();

choose = scan.nextInt();

**switch** (choose) {

**case** 1:

**if** (str1 != **null**){

**if** (ArgsCMD.*DBG*)

System.***out***.println("You choose 1. But string has been already created");

System.***out***.println("Error! String has been already created.");

**break**;

}

**else** {

**if**(ArgsCMD.*DBG*)

System.***out***.println("You choose 1. Creating string");

str1 = *InitializeString*();

**break**;

}

**case** 2:

**if** (str1 == **null**){

**if**(ArgsCMD.*DBG*) {

System.***out***.println("You choose 2. But your string is empty");

}

System.***out***.println("Error! String is empty!");

**break**;

}

**else** {

**if** (ArgsCMD.*DBG*) {

System.***out***.println("You choose 2. Showing your string");

}

*ShowString*(str1);

**break**;

}

**case** 3:

**if** (str1 == **null**){

**if**(ArgsCMD.*DBG*)

System.***out***.println("You choose 3. But your string is empty");

System.***out***.println("Error! String is empty!");

**break**;

}

**else** {

**if** (ArgsCMD.*DBG*)

System.***out***.println("You choose 3. Split text");

System.***out***.println("\nSplit on sentences:");

str2 = *CutText*(str1);

str1 = *CutText2*(str1);

System.***out***.println("Successfully!");

**break**;

}

**case** 4:

**if** (str1 == **null**){

**if** (ArgsCMD.*DBG*)

System.***out***.println("You choose 4. But your string is empty");

System.***out***.println("Error! String is empty!");

**break**;

}

**else** {

**if** (ArgsCMD.*DBG*)

System.***out***.println("You choose 4. Counting now");

System.***out***.println("\nEnter 1 for first or 2 for second: ");

**int** choice = scan.nextInt();

System.***out***.println("\nCount: ");

**while**(**true**) {

**if**(choice == 1) {

*Alphabet*(str1);

**break**;

}

**else** **if**(choice == 2) {

*Alphabet*(str2);

**break**;

}

**else**

System.***out***.println("Error. Repeat choose: ");

}

System.***out***.println("Successfully!");

**break**;

}

**case** 0:

**if** (ArgsCMD.*DBG*)

System.***out***.println("You choose 0. End of Work");

System.***out***.println("\n\nThanks for working!");

loop = **false**;

**break**;

**default**:

System.***out***.println("Error. Mistake number!");

**break**;

}

}

}

}

ArgsCMD.java

**package** ua.khpi.oop.Lunhol04;

/\*

\* class in which CMD commands are processed

\* \*/

**public** **class** ArgsCMD {

/\*\*

\* **@author** ALEXANDRA

\* **@version** 4.1

\* Here start point of the program

\* \*/

**static** **boolean** *DBG* = **false**;

**private** String[] args;

ArgsCMD(String[] args) {

**this**.args = args;

}

**boolean** isEmpty(){

**return** args.length == 0;

}

/\*

\* Function for comparing received commands

\* \*/

**void** arguments(){

**for** (String i: args){

**if** (i.equals("-d") || i.equals("-debug"))

ArgsCMD.*debug*();

**else** **if**(i.equals("-h") || i.equals("-help"))

ArgsCMD.*help*();

**else**

System.***out***.println("Error. Use only this arguments: \"-d\", \"-debug\", \"-h\", \"-help\"");

}

}

/\*

\* Function that displays information about yourself and about task when user calling the help

\* \*/

**private** **static** **void** help() {

System.***out***.println("Author: Lunhol Alexandra Sergeevna");

System.***out***.println("Individual tasks:");

System.***out***.println("Using the program of solving the problem of laboratory work #3, "

+ "\naccording to the application task provide the processing of user commands in the form of a text menu:\n" +

"data entry;\n" +

"view data;\n" +

"view data;\n" +

"displaying the result;\n" +

"program completion, etc.");

}

/\*

\* Function that displays additional information when you called debug

\* \*/

**private** **static** **void** debug() {

*DBG* = **true**;

System.***out***.println("Debug");

}

}

Function.java

**package** ua.khpi.oop.Lunhol04;

**import** java.util.Scanner;

/\*

\* class Function in which all used functions are stored

\*

\* @author ALEXANDRA

\* @version 4.2

\* Here start point of the program

\* \*/

**public** **class** Function {

/\*

\* Function that show user point of menu that he can choose

\* \*/

**static** **void** ChooseMenu() {

System.***out***.println("\n\nChoose the right variant:");

System.***out***.println("1. Create string");

System.***out***.println("2. Show string");

System.***out***.println("3. Split text");

System.***out***.println("4. Counting now");

System.***out***.println("0. The End");

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

}

/\*

\* Function show our string that we wrote

\* @param str is our string

\* \*/

**public** **static** **void** ShowString(String str){

System.***out***.println("Your string:");

System.***out***.println(str);

}

/\*

\* function alphabet counting vowel and other

\* @param v our vowels

\* \*/

**public** **static** **void** Alphabet (String str) {

**int** v = 0;

**int** sign = 0;

**char**[] vowels = {'a','e','i','o','u'};

**char** [] sentence = str.toLowerCase().toCharArray();

**for** (**char** letter : sentence) {

**if**(letter == ' ' || letter == '.' || letter == ',')

sign++;

**for** (**char** vowel : vowels) {

**if** (letter == vowel) {

v++;

}

}

}

System.***out***.println(str);

System.***out***.print("\tVowels:"+ v);

System.***out***.print("\tConsonants:" + (sentence.length -v-sign));

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

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

}

/\*

\* Function reads input string

\* @return str1 as our used string

\* \*/

**public** **static** String InitializeString(){

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

System.***out***.print("Enter text whith dots: ");

String str1 = **new** String();

str1 = scan.nextLine();

**return** str1;

}

/\*

\* Function truncates a string to a point and show the first sentence

\* @param index1 looking for dot

\* @return str

\* \*/

**public** **static** String CutText (String str) {

**int** index1=str.indexOf('.');

str = str.substring(index1+1);

**return** str;

}

/\*

\* Function truncates a string to a point and show the second sentence

\* @param index1 looking for dot

\* @return str

\* \*/

**public** **static** String CutText2 (String str) {

**int** index1=str.indexOf('.');

str = str.substring(0, index1+1);

**return** str;

}

}

РОБОТА ПРОГРАМИ

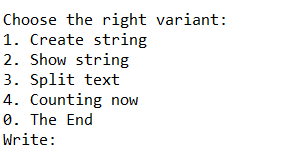


Рисунок 2 – робота програми

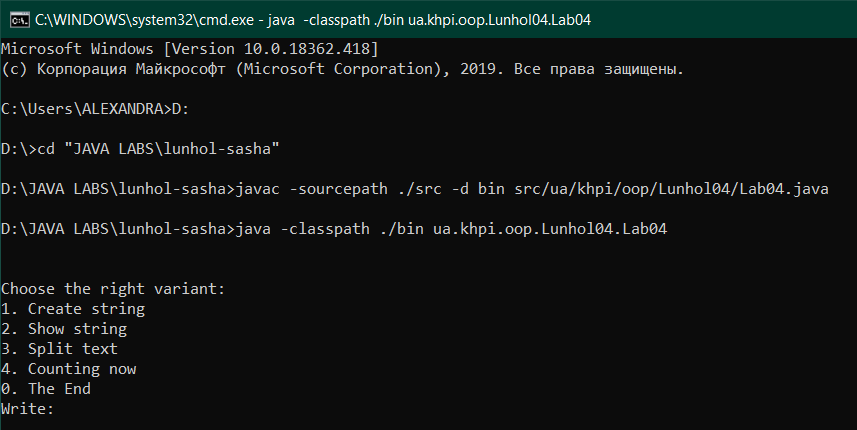


Рисунок 3 – Робота програми в командному рядку

ВИСНОВОК

В даній лабораторній роботі ми навчились запускати програму через командний рядок та виконувати дії в ньому а також прописувати основні запити типу help та debug.