Розробник:

Огонькова Наталія Максимівна

Група КІТ 101.8б

Варіант №11

**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, відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню:

* введення даних;
* перегляд даних;
* виконання обчислень;
* відображення результату;
* завершення програми і т.д.

1. Забезпечити обробку параметрів командного рядка 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.

1. ОПИС ПРОГРАМИ
   1. Опис змінних

String str3 = null; //слово, що вводить користувач

**int** word = 0; //змінна для підрахунку кількості слів

**int** begin\_of\_word = 0; //початок

**int** number = 5; //розмір слова, яке змінеться

String str1 = null; //текст, що вводить користувач

String str2; //змінна для тексту

**int** count = 0; //кількість відступів

1.2 Ієрархія та структура класів

**class** Lab04 – точка входу в програму

**class** Func – клас, що містить функції

**class** ArgsCMD – клас, в якому виконуються CMD команди

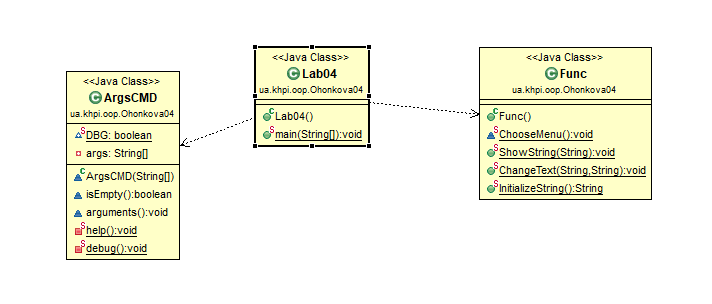


Рисунок 1 – Класи програми

* 1. Текст програми

**class** Lab04

package ua.khpi.oop.Ohonkova04;

import java.util.Arrays;

import java.util.Scanner;

import static ua.khpi.oop.Ohonkova04.Func.\*;

public class Lab04 {

/\*\*

\* @author NATALIA

\* @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 str3 = null;

while (loop) {

ChooseMenu();

choose = scan.nextInt();

switch (choose) {

case 1:

if (str1 != null){

if (ArgsCMD.DBG)

System.out.println("Error! Text shouldn`t be created.");

break;

}

else {

if(ArgsCMD.DBG)

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

System.out.print("Enter text: ");

str1 = InitializeString();

break;

}

case 2:

if (str1 == null){

if(ArgsCMD.DBG)

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

break;

}

else {

if (ArgsCMD.DBG) {

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

}

ShowString(str1);

break;

}

case 3:

if (str1 == null){

if(ArgsCMD.DBG)

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

break;

}

else {

if (ArgsCMD.DBG)

System.out.println("You choose 3. Input word");

System.out.println("\nEnter the word:");

str3 = InitializeString();

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

break;

}

case 4:

if (str1 == null){

if (ArgsCMD.DBG)

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

break;

}

else {

if (ArgsCMD.DBG)

System.out.println("You choose 4. Showing your new text");

ChangeText(str1,str3);

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;

}

}

}

}

**class** Func

**package** ua.khpi.oop.Ohonkova04;

**import** java.util.Scanner;

/\*

\* class Function in which all used functions are stored

\*

\* @author NATALIA

\* @version 4.2

\* Here start point of the program

\* \*/

**public** **class** Func {

/\*

\* 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 text");

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

System.***out***.println("3. Input word");

System.***out***.println("4. Show new text");

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** ChangeText (String str1, String str3) {

**int** word = 0;

**int** begin\_of\_word = 0;

**int** count = 0;

**int** number = 5;

String str2 = **null**;

**for** (**int** i = 0; i < str1.length(); i++) {

**if** (str1.charAt(i) == ' ') {

count++;

}

}

String[]wordArr = **new** String[++count];

**for** (**int** i = 0; i<str1.length();i++) {

**if**(str1.charAt(i)==' ') {

wordArr[word] = str1.substring(begin\_of\_word,i);

word++;

begin\_of\_word = i+1;

}

}

wordArr[word] = str1.substring(begin\_of\_word);

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

str2 = wordArr[i];

**if** (str2.length() == number) {

wordArr[i] = str3;

}

}

StringBuilder strBuild = **new** StringBuilder();

**for** (String s : wordArr) {

strBuild.append(s);

strBuild.append(" ");

}

str1 = strBuild.toString();

System.***out***.println(str1);

}

/\*

\* Function reads input string

\* @return str1 as our used string

\* \*/

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

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

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

str1 = scan.nextLine();

**return** str1;

}

}

**class** ArgsCMD

**package** ua.khpi.oop.Ohonkova04;

/\*

\* class in which CMD commands are processed

\* \*/

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

/\*\*

\* **@author** NATALIA

\* **@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: Ohonkova Natalia Maksymovna");

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");

}

}

1. ВАРІАНТИ ВИКОРИСТАННЯ

Програму можна використовувати для зміни слова заданої довжини в тексті на слово, яке задає користувач.

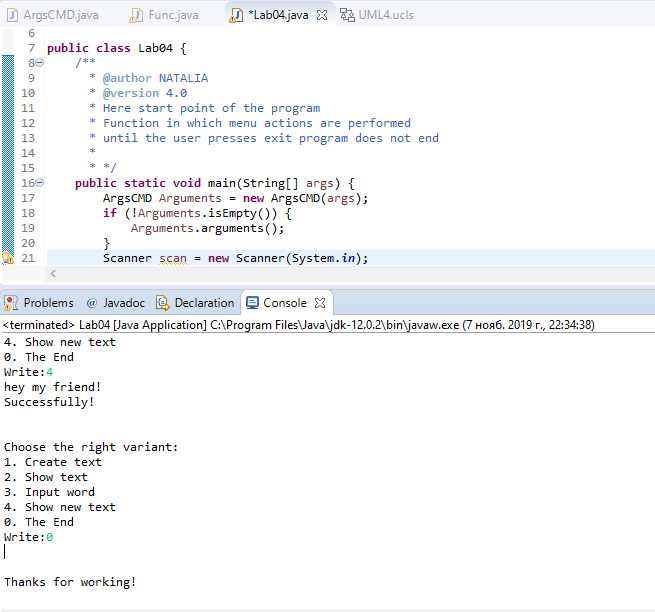


Рисунок 2 – Результат програми

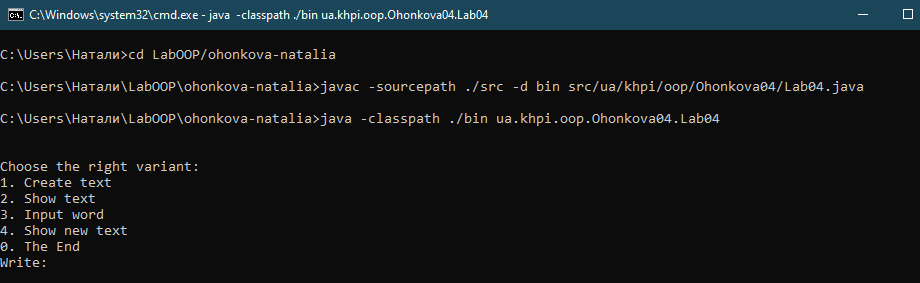


Рисунок 3 – Результат програми через команди CMD

ВИСНОВОК

При виконанні лабораторної роботи набуто практичних навичок, щодо розробки власних утилітарних класів та задач з використанням масивів і рядків за допомогою JDK платформи Java SE, а також реалізації діалогового режиму роботи з користувачем в консольних програмах мовою Java. Програма виконується без помилок.