Звіт

Автор: Пумня О., КІТ101.8б

Дата: 06.09.2019

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, 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.
2. 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.

ЗАВДАННЯ ДО РОБОТИ

Переробити попередню роботу так, щоб вона відповідала вимогам, описаним вище.

ОПИС ПРОГРАМИ

**Опис змінних**

String text; // текст, у якому проводиться пошук

String word; // слово, яке шукають

String sentence; // речення, яке буде додано після слова

String result; // текст – результат

ArgsHandler handler; // обробник консольних команд (-h, -d)

**Ієрархія та структура класів**

**class** Pumnya04 – точка входу в програму.

**class** UI – діалоговий клас, реалізує текстове меню роботи.

**class** StrBuilder – клас, який виконує індивідуальне завдання.

**class** ArgsHandler – клас, який обробляє консольні команди.

ТЕКСТ ПРОГРАМИ

**Текст файлу Pumnya04.java**

package pumnya04;

public class Pumnya04 {

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

ArgsHandler handler = new ArgsHandler(args);

if (!handler.empty()) {

handler.execute();

}

String text = null, word = null, sentence = null, result = null;

do {

UI.mainMenu();

UI.getChoice();

switch (UI.choice) {

case 0:

if(ArgsHandler.debug) {

System.out.println("\nYou chosen 0. Exiting...");

}

break;

case 1:

if(ArgsHandler.debug) {

System.out.println("\nYou chosen 1. Setting values...");

}

text = UI.getText();

word = UI.getWord();

sentence = UI.getSentence();

break;

case 2:

if(ArgsHandler.debug) {

System.out.println("\nYou chosen 2. Getting values...");

}

if (text != null && word != null && sentence != null) {

UI.printInfo(text, word, sentence);

}

else {

System.out.format("%nFirst you need to add values.");

}

break;

case 3:

if(ArgsHandler.debug) {

System.out.println("\nYou chosen 3. Executing task...");

}

if (text != null && word != null && sentence != null) {

result = StrBuilder.execute(text, word, sentence);

}

else {

System.out.format("%nFirst you need to add values.");

}

break;

case 4:

if(ArgsHandler.debug) {

System.out.println("\nYou chosen 4. Printing out result...");

}

if (text != null && word != null && sentence != null) {

UI.printResult(text, result);

}

else {

System.out.format("%nFirst you need to add values.");

}

break;

default:

System.out.println("\nEnter correct number.");

}

} while (UI.choice != 0);

}

}

**Текст файлу UI.java**

package pumnya04;

import java.util.Scanner;

class UI {

static int choice;

private static Scanner scan = new Scanner(System.in);

static void mainMenu() {

System.out.format("%n1. Enter values.%n");

System.out.format("2. Print values.%n");

System.out.format("3. Task completion.%n");

System.out.format("4. Print result.%n");

System.out.format("0. Exit.%n");

System.out.format("Enter your choose: ");

}

static void printInfo(final String text, final String word, final String sentence) {

System.out.format("%nText: " + text + "%n");

System.out.format("Word: " + word + "%n");

System.out.format("Sentence: " + sentence + "%n");

}

static void printResult(final String text, final String result) {

System.out.format("%nSource text: " + text + "%n");

System.out.format("Result: " + result + "%n");

}

static void getChoice() {

choice = scan.nextInt();

scan.nextLine();

}

static String getText() {

System.out.format("%nEnter text: ");

return scan.nextLine();

}

static String getWord() {

System.out.format("Enter word: ");

return scan.nextLine();

}

static String getSentence() {

System.out.format("Enter sentence: ");

return scan.nextLine();

}

}

**Текст файлу StrBuilder.java**

package pumnya04;

class StrBuilder {

static String execute(String text, String word, String sentence) {

if (ArgsHandler.debug) {

System.out.println("\nText: " + text);

System.out.println("Word: " + word);

System.out.println("Sentence: " + sentence);

}

StringBuilder builder = new StringBuilder(text);

int index = 0;

while (true) {

index = builder.indexOf(word, index);

if (index < 0)

break;

builder.insert(index + word.length(), sentence);

index += word.length();

}

return builder.toString();

}

}

**Текст файлу ArgsHandler.java**

package pumnya04;

class ArgsHandler {

private String[] args;

static boolean debug = false;

ArgsHandler(String[] args) { this.args = args; }

boolean empty() { return args.length == 0; }

void execute() {

for (String i : args) {

switch (i) {

case "-h":

System.out.println("\nAuthor: Pumnya Alexander, KIT101.8B");

System.out.println("Task: Enter text. Insert the specified word after each " +

"specified word. Output the initial text and result.");

System.out.println("-h (-help) : Print info about the work and console commands.");

System.out.println("-d (-debug) : Displays additional data.");

break;

case "-help":

System.out.println("\nAuthor: Pumnya Alexander, KIT101.8B.");

System.out.println("Task: Enter text. Insert the specified word after each " +

"specified word. Output the initial text and result.");

System.out.println("-h (-help) : Print info about the work and console commands.");

System.out.println("-d (-debug) : Displays additional data.");

break;

case "-d":

debug = true;

break;

case "-debug":

debug = true;

break;

default:

System.out.format("%n Incorrect command %s.%n", i);

System.out.println("-h, -help, -d, -debug is only allowed.");

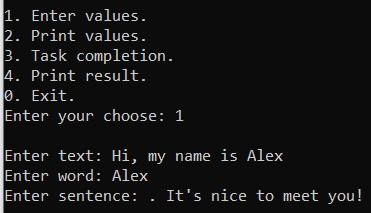
}

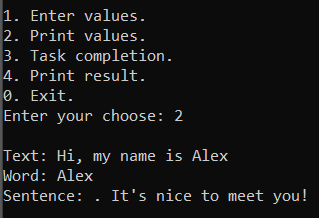
}

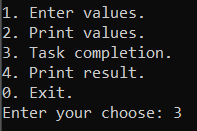
}

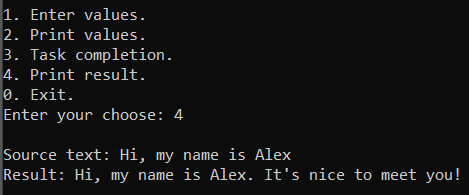
}

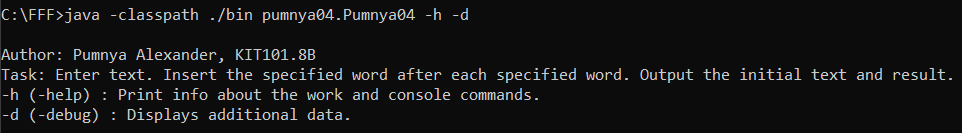
ВАРІАНТИ ВИКОРИСТАННЯ

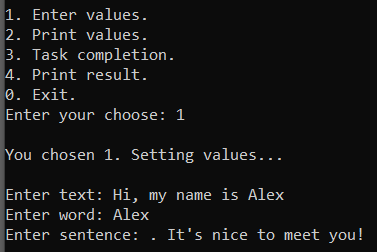
  
Рисунок 1 – Виконання першого пункту

  
Рисунок 2 – Виконання другого пункту

  
Рисунок 3 – Виконання третього пункту

  
Рисунок 4 – Виконання четвертого пункту

  
Рисунок 5 – Використання консольної команди -h

  
Рисунок 6 – Використання консольної команди -d

Програма може використовуватись як засіб для форматування текстів.

ВИСНОВКИ

При виконанні лабораторної роботи набуто практичних навичок щодо реалізації діалогового режиму роботи з користувачем в консольних програмах. Розроблено клас, що реалізує діалоговий режим роботи, а також клас, який оброблює консольні команди.