Тема: Інтерактивні консольні програми для платформи Java SE.

**Мета:** Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою Java.

**1 ВИМОГИ  
1.1 Розробник**

Інформація про розробника:

- Когутенко Олександр Олексійович;

- КІТ-119Д;

- 11 варіант.

**1.2 Загальне завдання**

1.Використовуючи програму рішення завдання [лабораторної роботи №3](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/), відповідно до [прикладної задачі](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/" \l "_4) 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)" \l ".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.

1. 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.

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Використовуеться інкапсуляція у классі HelperClassWithСonsole.

2.2 Ієрархія та структура класів

Маємо два класи: Main та HelperClassWithСonsole.

У Main виконується інтерактив з користувачем. Тобто форматируваний вивід та корректне прийняття данних з використанням прийняття параметрів що потрібні за прикладною задачею, однак добавивши пару своїх.

У HelperClassWithСonsole виконуються основні дії за прикладною задачею. Тобто вивід допоміжної інформації, параметр який вказує значення змінніх та прикладна задача з [лабораторної роботи №3](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task03/).

2.3 Важливі фрагменти програми

package ua.khpi.oop.kogutenko04;  
  
import ua.khpi.oop.kogutenko03.HelperClassWithString;  
  
import java.util.Scanner;  
public class Main {  
  
 */\*\*  
 \* The entry point of application.  
 \*  
 \** ***@param*** *args the input arguments  
 \*/* public static void main(String[] args)  
 {  
 *start\_*();  
 }  
  
 private static void start\_()  
 {  
 boolean check = true, checkHelpLine = true;  
 String input, nikname;  
 HelperClassWithConsole helper = new HelperClassWithConsole();  
 Scanner scanner = new Scanner(System.*in*);  
 try  
 {  
 System.*out*.print("Input your nikname: ");  
 nikname = scanner.nextLine();  
 //clearConsole();  
 while (check)  
 {  
 if (checkHelpLine) {  
 System.*out*.println("Hello, my name is Alex Kogutenko\n"  
 + "I am from Ukrain and studing at NTU \"KHPI\"\n"  
 + "This is a test console-project with a debug programs.\n"  
 + "Such commands are present so far:\n"  
 + "\t-h | -help \t-\t command for summary information about other commands (important to remmember!)\n"  
 + "\t-d | -debug \t-\t file debugger command\n"  
 + "\tchtext\t-\t changed the text as in the past lab work (lab work 3)\n"  
 + "\texit \t-\t exit form program\n");  
 checkHelpLine = false;  
 }  
 System.*out*.print(nikname + "@" + nikname + ": ");  
 input = scanner.nextLine();  
 switch (input) {  
 case " ": {  
 break;  
 }  
 case "-h": {  
 helper.printHelpInfo();  
 break;  
 }  
 case "-help": {  
 helper.printHelpInfo();  
 break;  
 }  
 case "-d": {  
 helper.debuggerInHelper();  
 break;  
 }  
 case "-debug": {  
 helper.debuggerInHelper();  
 break;  
 }  
 case "chtext": {  
 helper.changedText();  
 break;  
 }  
 case "exit":  
 {  
 check = false;  
 break;  
 }  
 default: {  
 System.*out*.println("(" + input + ") I don't know this command :(");  
 break;  
 }  
 }  
 }  
 System.*out*.println("GOOD BEY!!!");  
 } catch (Exception e) {  
 System.*out*.println(e);  
 check = false;  
 }  
 }  
}

package ua.khpi.oop.kogutenko04;  
  
import ua.khpi.oop.kogutenko03.HelperClassWithString;  
  
import java.lang.StringBuilder;  
import java.text.SimpleDateFormat;  
import java.util.Collections;  
import java.util.Date;  
import java.util.Scanner;  
  
*/\*\*  
 \* The type Helper class with console.  
 \*/*public class HelperClassWithConsole {  
  
 private StringBuilder str;  
 private Integer count;  
 private Integer length;  
…

public String getInfoOfHelperObject()  
 {  
 return str.getClass().getSimpleName() + " 'str' has '" + getStr() + "'\n"  
 + count.getClass().getSimpleName() + " 'count' has '" + getCount() + "'\n"  
 + length.getClass().getSimpleName() + " 'length' has '" + getLength() + "'";  
 }  
  
 public void setInfoHelperObject(String str)  
 {  
 setStr(str);  
 setCount(CountWordsInHelper());  
 setLength(str.length());  
 }  
 …

*/\*\*  
 \* Print help info.  
 \*/* public void printHelpInfo() {  
 System.*out*.println("Hello, my name is Alex Kogutenko\n" + "I am from Ukrain and studing at NTU \"KHPI\"\n"  
 + "This is a test console-project with a debug programs.\n" + "Such commands are present so far:\n"  
 + "\t-h | -help \t-\t command for summary information about other commands (important to remmember!)\n"  
 + "\t-d | -debug \t-\t file debugger command\n" + "\t-clear \t-\t cleaning the console\n");  
 System.*out*.println("Tap any key...");  
 }  
  
 */\*\*  
 \* Debugger in helper.  
 \*/* public void debuggerInHelper()  
 {  
 Date date = new Date();  
 SimpleDateFormat formatDate = new SimpleDateFormat("'\nDate:' dd.MM.yyyy '\nTime: 'HH:mm:ss '\n'" );  
 System.*out*.println(formatDate.format(date) + "values of the variables:\n"  
 + getInfoOfHelperObject());  
  
 }

public void changedText()  
 {  
 System.*out*.println();  
 System.*out*.println("Enter the text. In the text,\nreplace the words of the specified length with the specified line");  
 Scanner scan = new Scanner(System.*in*);  
 System.*out*.print("Main line: ");  
 String mainStr = scan.nextLine();  
 setInfoHelperObject(mainStr);  
 System.*out*.print("Enter number of letters in word which you want to changed: ");  
 int length = scan.nextInt();  
 System.*out*.print("Enter word to replace: ");  
 scan.nextLine();  
 String newWord = scan.nextLine();  
 String newStr = replaceAllWordsOn(length, newWord);//new String(mainHelperStr.replaceAllWordsOn(length, newWord));  
 System.*out*.println("----------------------------------------------");  
 System.*out*.println("No changed line: " + mainStr);  
 System.*out*.println("Result: " + newStr);  
 //scan.close();  
 }  
  
}

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

Програма предназначена за для розшириного використання лабороторної роботи №3.

**ВИСНОВКИ**

Розробив інтерактивну консольну програму для платформи Java SE.

Ознайомився з реалізацією діалогового режиму роботи з користувачем в консольних програмах мовою Java.