**Лабораторна робота №4**

**Інтерактивні консольні програми для платформи Java SE**

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

**1 ВИМОГИ**

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)#.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.

**1.1 Розробник**

***Виконавець:*** Сиромятников Марк

***Група:*** КІТ-119а

**Варіант:** 19

**2 ОПИС ПРОГРАМИ**

Було використано наступні засоби:

* *import java.util.Scanner* – для вводу даних.
* System.***out***.println() – для виводу тексту у консоль.
* .close(); - закриття потоку вводу
* StringBuilder – для викликання методів виконання завдання.

**2.1 Ієрархія та структура класів**

Було створено 1 клас MainClass, який викликає метод виконання завдання та клас-хелпер до нього HelperClass, містить у собі метод виконання завдання. Розроблено меню у класі MainClass, що активує за бажанням користувача відповідні методи.

**2.2 Важливі фрагменти програми**

while(tryIt) {

System.out.println("1. Enter data.");

System.out.println("2. Show current.");

System.out.println("3. Do task (Replace every k`th symbol of the word with entered or by default).");

System.out.println("4. Show result in table.");

System.out.println("5. Close menu.");

System.out.println("Chose number of option : ");

Scanner sc = new Scanner(System.in);

int answer = sc.nextInt();

switch(answer) {

case 1:

Scanner sc1 = new Scanner(System.in);

in.delete(0, in.length());

System.out.print("Input string: ");

in.append(sc1.nextLine());

System.out.print("Input symbol to replace: ");

chaReplace= sc1.nextLine();

System.out.print("Input position: ");

pos = sc1.nextInt();

StringBuffer text = new StringBuffer(in);

if(pos > text.length())

{

System.out.println("Error: Position is greater then length of string.");

sc1.close();

return;

}

break;

case 2:

System.out.println();

System.out.println("\nCurrent string is : " + in);

System.out.println("\nCurrent position is : " + pos);

System.out.println("\nCurrent symbol is : " + chaReplace);

System.out.println(" ");

break;

case 3:

in = HelperClass.replaceInPosition(in, cnt, chaReplace, pos);

break;

case 4:

HelperClass.outputData(in, str, cnt, chaReplace, pos);

break;

case 5:

tryIt = false;

System.out.println("End\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

break;

default:

System.out.println("Error: Out of range of variation. Try one more time. \n");

};

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

Нижче наведено опис пунктів меню та прикріплено відповідні зображення задля пояснення роботи. Пункти меню, відповідно, можуть бути викликані користувачем

1. Пункт меню “Enter data.” Викликає метод обробки введеного користувачем тексту задля виконання завдання з попередньої лабораторної (рис 1).

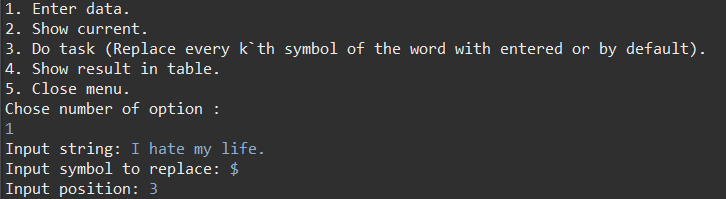


Рисунок 1 - Перший пункт меню

1. Пункт меню “Show current” відображає поточні дані, які будуть оброблятися методами. За для полегшення роботи у режимі «дебаг» дані вже введено замість користувача (рис 2).

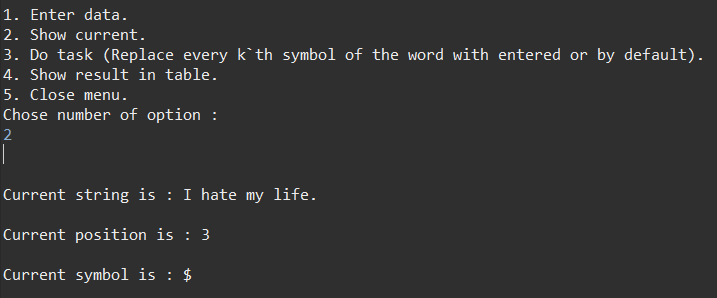


Рисунок 2 - Другий пункт меню

1. Пункт меню “Do task” викликає метод, розроблений у попередній лабораторній роботі стосовно поточних даних (рис 3).

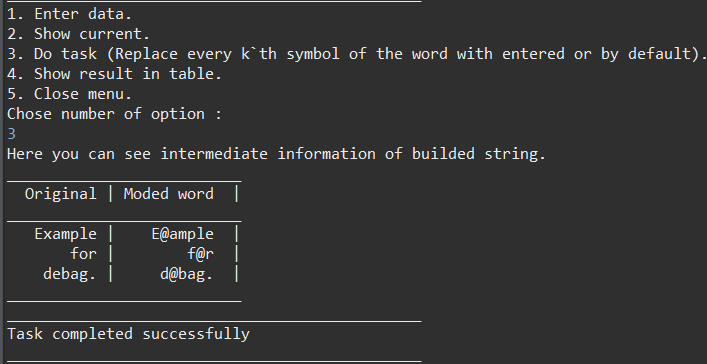


Рисунок 3 - Третій пункт меню

1. Пункт меню “Show result in table” відображає виконане завдання, якщо його виконував користувач у вигляді таблиці з усіма необхідними даними.

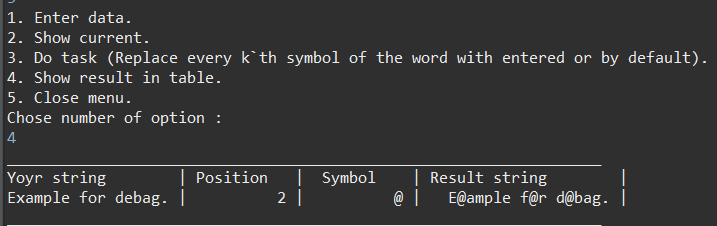


Рисунок 4 - Четвертий пункт меню

1. Пункт меню “Close menu” зупиняє роботу програми та виходить з неї, видаючи на екран повідомлення.

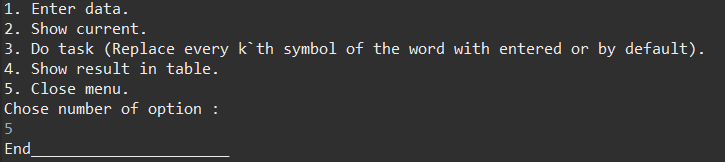


Рисунок 5 – П’ятий пункт меню

1. Програма може запускатися за допомогою консольних аргументів у режимі –h або –help та у режими –d або –debug. При виконанні програми у першому режимі виводяться дані про її виконавця (рис 6). Виконання у другому режимі продемонстровано на попередніх рисунках.

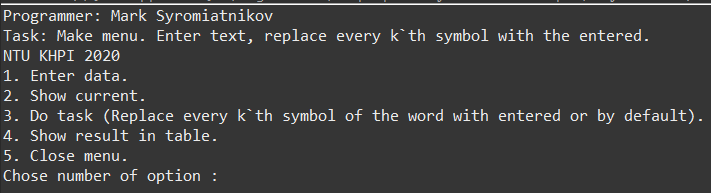


Рисунок 6 – Виконання у режимі -h

**ВИСНОВКИ**

У результаті виконання лабораторної роботи було створено власний утилітарний клас, розроблено програму обробки масивів рядків.