**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, 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:
   * 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. **Розробник**

* П.І.Б: Заночкин. Є. Д.
* Группа: КІТ-119а
* Варіант: 7
  1. **Загальне завдання**

Ввести текст. З тексту видалити всі слова заданої довжини, що починаються на приголосну літеру. Вивести початковий текст та результат.

1. **ОПИС ПРОГРАМИ**
   1. **Було використано наступні засоби:**

StringBuilder sbStr = new StringBuilder(str.substring(0)) – створення рядку типу StringBuilder;

sbStr.length() – визначення довжини рядка;

sbStr.charAt() – визначення символу, який стоїть на заданому місці.

* 1. **Ієрархія та структура класів**

Було створено 2 класи:

* public class Main – містить метод main, та метод Menu, в якому реалізовано діалоговий режим роботи з користувачем;
* public class DeleteWords – клас для вирішення загального завдання.
  1. **Важливі фрагменти програми**

public static void Menu(boolean debug)

{

String str = new String("There are five types of schools in the US educational system. They are: kindergarten, elementary school, middle school, high school and private school. Children go to kindergarten when they are 5 years old. They go to elementary school from ages 6 through 11 (1-5 grades), middle school from ages 12 through 14 (6-8 grades) and high school from ages 15 through 19 (9-12 grades).");

boolean endCheck = true;

int count = 0;

StringBuilder sb = new StringBuilder(str);

Scanner in = new Scanner(System.in);

while (endCheck)

{

System.out.println("1. Enter new text");

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

System.out.println("3. Delete words that start with a consonant");

System.out.println("4. Exit");

System.out.println("Enter your option:");

int option = in.nextInt();

switch (option)

{

case 1:

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

System.out.println("\nEnter new text: ");

sb.append(in.nextLine());

sb.append(in.nextLine());

System.out.println();

break;

case 2:

System.out.println("\n" + sb + "\n");

break;

case 3:

System.out.println("\nEnter count of digit in word to delete:");

count = in.nextInt();

DeleteWords.findAndDelete(sb.toString(), count, debug);

System.out.println();

break;

case 4:

endCheck = false;

in.close();

System.out.println("\nEnd of work");

break;

default:

System.out.println("\nWrong command\n");

break;

}

}

}

public class DeleteWords

{

/\*\*

\* Метод findAndDelete, який знаходить та видаляє потрібні слова.

\*/

public static void findAndDelete(String str, int count, boolean debug)

{

StringBuilder sbStr = new StringBuilder(str.substring(0));

System.out.println("\nDefault string: " + sbStr);

int endPos = 0;

if(debug)

System.out.println("\nCurrent text: " + sbStr + "\n");

for (int i = 0; i < sbStr.length(); i++)

{

endPos = findPunctMark(sbStr,i);

if(!Character.isDigit(sbStr.charAt(i)))

{

if(Character.isAlphabetic(sbStr.charAt(i)))

{

if (isVowel(sbStr.charAt(i)) == false)

{

StringBuilder sb = new StringBuilder(sbStr.substring(i, endPos));

int endPos2 = findPunctMark(sb,0);

boolean flag = false;

if(endPos2 == -1)

flag = true;

while(endPos2 != -1)

{

sb.delete(endPos2,sb.length());

endPos--;

endPos2 = findPunctMark(sb,0);

}

if(debug)

System.out.println("Current word: " + sb);

if(endPos == sbStr.length()-1)

if(sb.length() == count)

{

sbStr.delete(i, endPos);

i = i - 1;

}

else

i = endPos;

else

if(sb.length() == count)

{

if(flag)

sbStr.delete(i, endPos+1);

else

sbStr.delete(i, endPos);

i = i - 1;

}

else

i = endPos;

}

else

i = endPos;

}

}

else

i = endPos;

if(debug)

System.out.println("Current text: " + sbStr);

}

System.out.println("\nEdited string: " + sbStr);

}

/\*\*

\* Метод isVowel, який перевіряє першу літеру слова.

\*/

public static boolean isVowel(char c)

{

switch (Character.toLowerCase(c))

{

case 'a':

case 'e':

case 'i':

case 'o':

case 'u':

case 'y':

return true;

default:

return false;

}

}

/\*\*

\* Метод findPunctMark, який знаходить позицію, на якій є пунктуаційний знак.

\*/

public static int findPunctMark(StringBuilder a, int num)

{

int endPos = a.indexOf(" ", num);

if(endPos == -1)

endPos = a.indexOf(".", num);

if(endPos == -1)

endPos = a.indexOf("!", num);

if(endPos == -1)

endPos = a.indexOf("?", num);

if(endPos == -1)

endPos = a.indexOf(",", num);

if(endPos == -1)

endPos = a.indexOf(";", num);

if(endPos == -1)

endPos = a.indexOf(":", num);

if(endPos == -1)

endPos = a.indexOf("-", num);

if(endPos == -1)

endPos = a.indexOf("\\", num);

if(endPos == -1)

endPos = a.indexOf("/", num);

if(endPos == -1)

endPos = a.indexOf("|", num);

if(endPos == -1)

endPos = a.indexOf(")", num);

if(endPos == -1)

endPos = a.indexOf("(", num);

if(endPos == -1)

endPos = a.indexOf("@", num);

if(endPos == -1)

endPos = a.indexOf("#", num);

if(endPos == -1)

endPos = a.indexOf("$", num);

if(endPos == -1)

endPos = a.indexOf("%", num);

return endPos;

}

}

1. **Варіанти використання**

Під час виконання програми, користувач може:

1. Ввести текст з клавіатури;
2. Вивести текст у консоль;
3. Виконати основне завдання;
4. Завершити виконання програми.

Перед запуском програми є можливість ввести додаткові параметри, такі як: -d чи –debug, які виводять додаткову інформацію про текст та операції, які виконуються над цим текстом та –h чи –help, які виводять інформацію про автора та завдання.

* + 1. **Результати роботи програми**

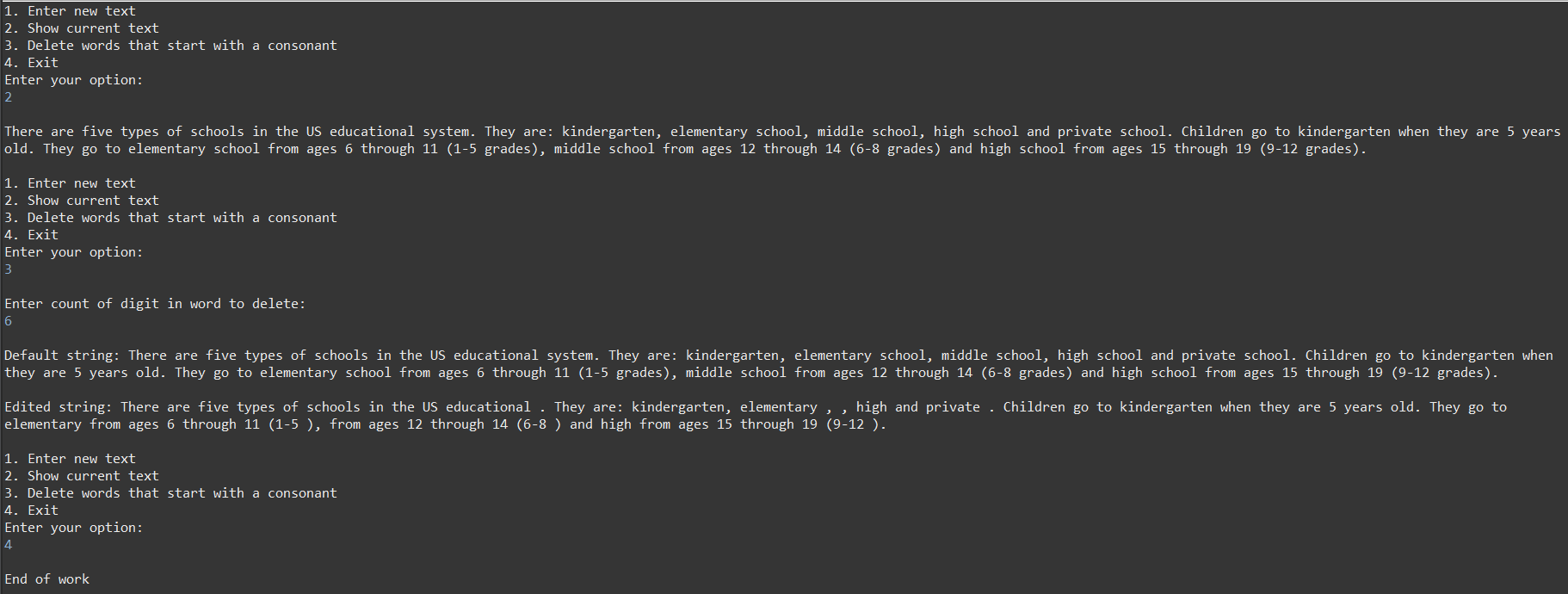
****

Рисунок 4.1 – Результат роботи програми у середовищі Eclipse

**Висновок**

Під час виконання лабораторної роботи було набуто навички роботи з інтерактивними консольними програмами у середовищі Eclipse IDE.