**Звіт**

**Лабораторна работа 4. Інтерактивні консольні програми для платформи Java SE**

**Мета роботи**: 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. **Розробник**: Капелька Ярослав Іванович, КІТ119-а, варіант №9.

1. **ОПИС ПРОГРАМИ**
   1. **Засоби ООП**: клас, метод класу, поле класу.
   2. **Ієрархія та структура класів:** один публічний клас Main з двома методами та клас HelperClass з класом util та одним методом.
   3. **Важливі фрагменти програми:**

**public** **class** HelpMenu

{

**public** **static** **class** util

{

**static** StringBuilder *builder* = **new** StringBuilder();

**static** java.util.Scanner *in* = **new** java.util.Scanner(System.***in***);

**static** java.util.HashSet<String> *myHashSet* = **new** java.util.HashSet<String>();

**static** String getString()

{

System.***out***.println("Введите текст: ");

String res = *in*.nextLine();

**return** res;

}

**static** **void** replacement(String text)

{

text += " ";

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

{

**if** (text.charAt(i) != ' ')

{

*builder*.append(text.charAt(i));

}

**else**

{

**if** (!*builder*.toString().equals(""))

{

*myHashSet*.add(*builder*.toString());

*builder*.setLength(0);

}

}

}

}

**static** **int** substringCount(String s, String pattern)

{

**int** result = 0;

s += " ";

pattern += " ";

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

{

**if**(i + pattern.length() <= s.length())

{

**if** (s.substring(i, i + pattern.length()).equals(pattern))

{

result++;

i += pattern.length()-1;

}

}

}

**return** result;

}

**public** **static** **void** task()

{

String data = *getString*();

*replacement*(data);

**int** res;

**for**(String s : *myHashSet*)

{

res = *substringCount*(data,s);

System.***out***.println("Слово: " + s);

System.***out***.println("Повторения: " + res);

}

}

**public** **static** **void** task(String data)

{

*replacement*(data);

**int** res;

**for**(String s : *myHashSet*)

{

res = *substringCount*(data,s);

System.***out***.println("Слово: " + s);

System.***out***.println("Повторения: " + res);

}

}

}

**private** **static** **boolean** *debug* = **false**;

**public** **static** **void** debugMod(**boolean** debugMod)

{

*debug* = debugMod;

System.***out***.printf("Debug mod ");

**if** (*debug*)

{

System.***out***.printf("on\n");

}

**else**

{

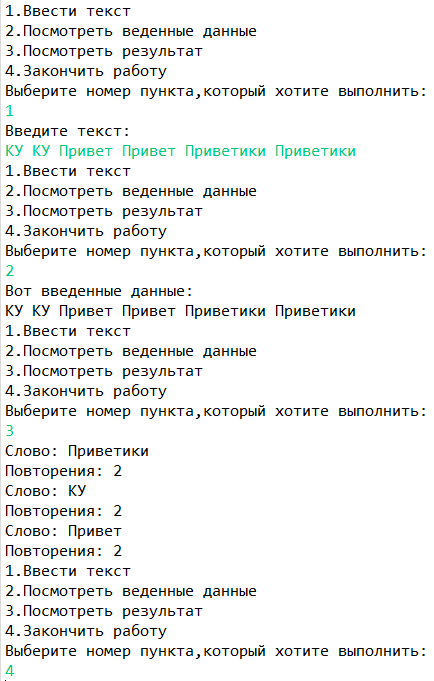
System.***out***.printf("off\n");

}

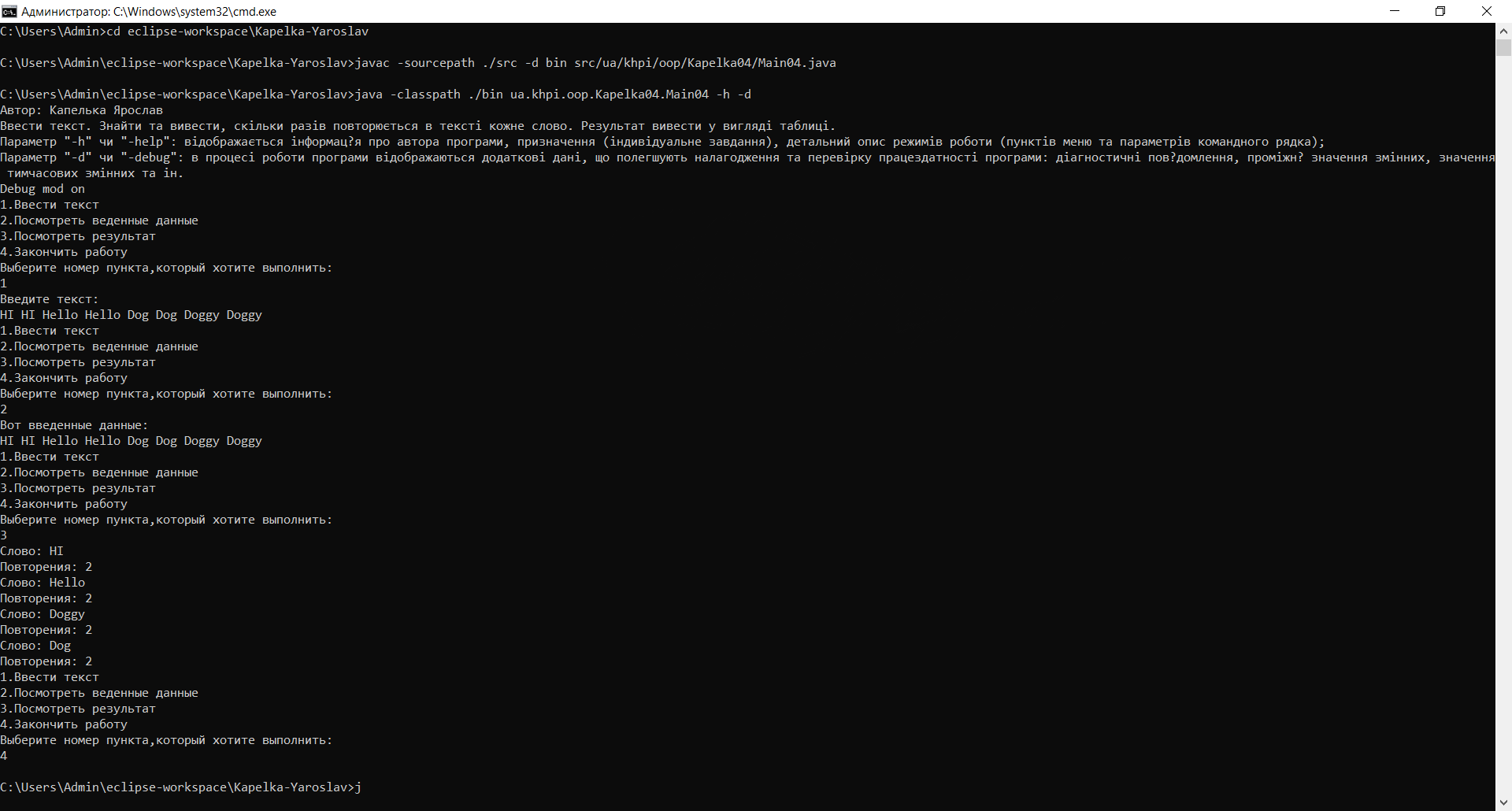
}

}

**Результат виконання програми:**

****

**Результат виконання консолі:**

****

**Висновки**

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з інтерактивними консольними програмами для платформи Java SE.

Програма протестована, виконується без помилок.