Лабораторна робота №1.

Мета:

● Розробка власних утилітарних класів.

● Набуття навичок вирішення прикладних задач з використанням масивів і рядків.

● Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою Java.

1. Вимоги

1. Розробити та продемонструвати консольну програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 15 зменшеного на одиницю номера студента в журналі групи.

2. Використовуючи програму рішення завдання відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню:

a. введення даних;
b. перегляд даних;
c. виконання обчислень;
d. відображення результату;
e. завершення програми і т.д.

3. Забезпечити обробку параметрів командного рядка для визначення режиму роботи програми:

a. параметр “-h” чи “-help”: відображається інформація про автора програми, призначення (індивідуальне завдання), детальний опис режимів роботи (пунктів меню та параметрів командного рядка);

b. 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.

4. При вирішенні прикладних задач використовувати латинку .

5. Продемонструвати використання об’єктів класу StringBuilder або StringBuffer .

6. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. Helper Class) та для обробки даних використовувати відповідні статичні методи.

7. Забороняється використовувати засоби обробки регулярних виразів: класи пакету java.util.regex (Pattern , Matcher та ін.), а також відповідні методи класу String (matches , replace , replaceFirst , replaceAll , split).

1.1 Розробник

Левицький Богдан, КН-108, номер варіанту – 14.

1.2 Задача

Ввести текст. Після кожного слова тексту, що закінчується заданим символом, вставити зазначений рядок. Вивести початковий текст та результат

2 Опис програми

Програма після зазначеного символа вставляє зазначений рядок в текст й виводить результат. Відповідно до завдання, добавлено меню, команди до якого викликаються за допомогою консольної команди -h(режим допомоги та інформації) та та режим debug –d(-debug), що виведе покрокову обробку тексту.

2.1 Засоби ООП

Для виконання завдання був використаний утилітарний клас EditString, який містив певні статичні поля і методи необхідні для редагування.

2.2 Ієрархія та структура класів

1. Клас FirstLab, який містить 1 функцію – main, тобто саме меню.

2. Клас `EditString` з статистичними полями та декількома функціями:

```
public class EditString
{
    static StringBuilder result = null;
    static boolean posted = false;
    static StringBuilder start(String inputString,char keySymbol,String keyString,boolean debug)
    {
        result = new StringBuilder();
        for(int i = 0;i < inputString.length();i++)
        {
            result.append(inputString.charAt(i));
            if(debug && (inputString.charAt(i) == ' ' || i == inputString.length()-1))
                System.out.println(result);
            if((i+1 < inputString.length() && inputString.charAt(i+1) == ' ') || i == inputString.length()-1
            {
                if(inputString.charAt(i) == keySymbol)
                {
                    result.append(keyString);
                    posted = true;
                }
            }
        }
        if(!posted)
        {
            System.out.println("Ключового символу в стрічці немає");
            result = null;
            return null;
        }
        return result;
    }
}
```

2.3 Важливі фрагменти програми.

```java
import java.util.Scanner;
public class FirstLab {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        boolean debug = false;
        for(int i = 0;i < args.length;i++)
        {
            if(args[i].equals("-h") || args[i].equals("-help"))
            {
                System.out.println(""
                        + "Виконав: Левицький Богдан, КН-108\n\n"
                        + "Мета: \n"
                        + "Ввести текст.\n"
                        + "Після кожного слова тексту, що закінчується заданим символом, вставити зазначений рядок.
                        + "Вивести початковий текст та результат.\n\n"
                        + "Команди меню:\n"
                        + "Ввести текст: -w\n"
                        + "Перегляд введених даних: -i\n"
                        + "Виконання обчислень: -s\n"
                        + "Вивести результат: -r\n"
                        + "Вийти: -e"
                        + "\n");
            }
            else if(args[i].equals("-d") || args[i].equals("debug"))
            {
                debug = true;
                System.out.println("Програма запущена в режимі debug.\n");
            }
        }
            int wrongCommand = 0;
            String command = null;
            String inputString = null;
```

```java
            char keySymbol = ' ';
            String keyString = null;
            Scanner in = new Scanner(System.in);
            System.out.print("Введіть команду:\r");
            command = in.nextLine();
            StringBuilder result = null;
            for(;;System.out.print("Введіть команду:\r"),command = in.nextLine())
            {
                if(command.length()-1 >= 1)
                switch(command.charAt(1))
                {
                    case'w':
                    {
                        wrongCommand = 0;
                        System.out.println("Введіть текст:\n");
                        inputString = in.nextLine();
                        System.out.println("Введіть ключовий символ: \n");
                        keySymbol = in.nextLine().charAt(0);
                        System.out.println("Введіть ключовий рядок: \n");
                        keyString = in.nextLine();
                        result = null;
                        EditString.result = null;
                        break;
                    }
                    case'i':
                    {
                        wrongCommand = 0;
                        System.out.print("Введені дані: \n"
                                + "Текст: " + inputString + "\n"
                                + "Ключовий симовл: " + keySymbol + "\n"
                                + "Ключовий рядок: " + keyString + "\n");
                        if(result != null)
                            System.out.print("Результат: " + result + "\n");
                        break;
```

```
                        System.out.print("Результат: " + result + "\n");
                    break;
                }
                case's':
                {
                    wrongCommand = 0;
                    result = EditString.start(inputString, keySymbol, keyString,debug);
                    break;
                }
                case'r':
                {
                    wrongCommand = 0;
                    if(result != null)
                    System.out.println("Результат: " + result);
                    break;
                }
                case'e':
                {
                    System.out.println("Завершення роботи програми..");
                    return;
                }
                default:
                {
                    System.out.println("Команду введено невірно.\n");
                    wrongCommand++;
                    if(wrongCommand == 3)
                    {
                        System.out.println("Запустіть програму з параметром -h для отримання допомоги.");
                        return;
                    }
                }
            }
        }
    }
```

3. Варіанти використання

Дана програма може використовуватись для редагування помилки допущені у тексті.

ВИСНОВКИ

У ході роботи ми розробили власний утилітарний клас, набули навичок використання StringBuilder, розробили діалоговий режим роботи з користувачем.