

УТИЛІТАРНІ КЛАСИ. ОБРОБКА МАСИВІВ І РЯДКІВ

Мета: Розробка власних утилітарних класів. Набуття навичок вирішення прикладних задач з використанням масивів і рядків.

ВИМОГИ

Розробник:

- Веремчук Дарина Анатоліївна;
- КІТ-119д;
- Варіант №5.

Загальне завдання:

- 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) При вирішенні прикладних задач використовувати латинку.
- 3) Пропридемонструвати використання об'єктів класу `StringBuilder` або `StringBuffer`.
- 4) Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. `Helper Class`) та для обробки даних використовувати відповідні статичні методи.
- 5) 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`).

Завдання до лабораторної роботи:

5. Ввести текст. У тексті кожну літеру замінити її номером в алфавіті. Вивести результат наступним чином: в одному рядку друкувати текст з двома пропусками між буквами, в наступному рядку під кожною буквою друкувати її номер.

ОПИС ПРОГРАМИ

Опис змінних:

Scanner in;	// змінна для введення даних
String text;	// текст, у якому проводиться пошук
final String[] lines;	// масив підрядків
List<String> words;	// список рядків

Ієрархія та структура класів:

class Main – точка входу в програму;

class Helper - клас, що виконує індивідуальне завдання.

ТЕКСТ ПРОГРАМИ

Текст класу **Main**:

```
package ua.oop.khpi.veremchuk03;

import java.util.Scanner;

public class Main {
```

```

    /**
     * An entry point - main method.
     *
     * @param args - arguments of main method
     */
    public static void main(String[] args) {
        System.out.println("Enter text:\n");
        Scanner in = new Scanner(System.in);
        String text = in.nextLine();
        final String[] lines = Helper.DivString(text); // alternative to split method
        for (final String line : lines) {
            Helper.printSymbols(line);
            Helper.printSymbolNumbers(line);
        }
    }
}

```

Текст класу **Helper**:

```

package ua.oop.khpi.veremchuk03;

import java.util.ArrayList;
import java.util.List;

public class Helper {
    /**
     * The method that printing symbols with using String Builder object
     *
     * @param line - our text after splitting
     */
}

```

```

    */

    public static void printSymbols(final String line) {

        StringBuilder builder = new StringBuilder();

        for (char symbol : line.toCharArray()) {

            //Check for the ' '/'!'/'.'/' ':'

            if((int)symbol == 32 | (int)symbol == 33 |(int)symbol == 46 |(int)symbol == 58
|(int)symbol == 44)

                {

                    continue;

                }

            builder.append("\t");

            builder.append(symbol);

        }

        System.out.print(builder.toString());

        System.out.println();

    }

    /**

    * The method that printing integer value (ASCII CODE) of symbol

    *

    * @param line - our line of symbols where we looking for (ASCII CODE)

    */

    public static void printSymbolNumbers(final String line) {

        StringBuilder builder = new StringBuilder();

        for (char symbol : line.toCharArray()) {

            //niSnïSnïSnïSnïSnïSnïS niSnïS ' '/'!'/'.'/' ':'

            if((int)symbol == 32 | (int)symbol == 33 |(int)symbol == 58|(int)symbol ==
44|(int)symbol == 46)

```

```

        {
            continue;
        }

        builder.append("\t");

        builder.append((int)symbol);

    }

    System.out.print(builder.toString());

    System.out.println();
}

/**
 * The method that splits text
 *
 * @param text - text we are splitting
 * @return - returns array of words
 */

public static String[] DivString(String text) {
    List<String> words = new ArrayList<>();

    StringBuilder builder = new StringBuilder();

    for(char symbol : text.toCharArray()) {
        if((int)symbol == 32 | (int)symbol == 33 |(int)symbol == 58|(int)symbol ==
44|(int)symbol == 46) {
            continue;
        }

        builder.append(symbol);
    }

    words.add(builder.toString());
}

```

```

        builder = new StringBuilder();

        if(builder.length() != 0) {

            words.add(builder.toString());

        }

        for (int i = 0; i < words.size(); i++) {

            if(words.get(i).length() == 0) {

                words.remove(i);

            }

        }

        String[] output = new String[words.size()];

        for (int i = 0; i < words.size(); i++) {

            output[i] = words.get(i);

        }

        return output;

    }
}

```

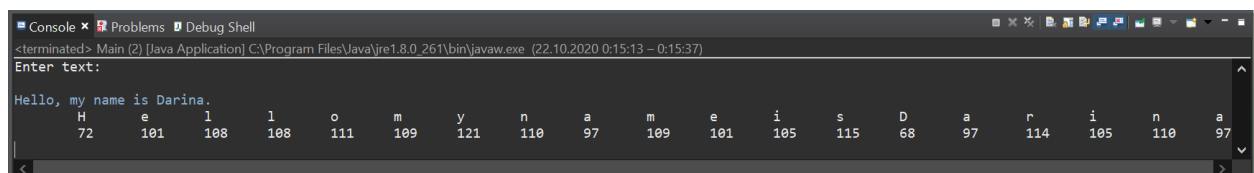


Рисунок 3.1 – Результат роботи програми

ВАРІАНТИ ВИКОРИСТАННЯ

Програму можна використовувати для заміни символів (десяткових цифр, латинського алфавіту, розділових знаків, керуючих символів) їх числовим кодом.

ВИСНОВОК

Отримала практичні навички розробки власних утилітарних класів та вирішення прикладних задач з використанням масивів і рядків. Написала утилітарний клас `Helper`, який використовує об'єкт класу `StringBuilder` та виконує індивідуальне завдання. Використала пакет `java.util.Scanner`, `import java.util.ArrayList`, `import java.util.List`. Програма виконується без помилок.