

ЛАБОРАТОРНА РОБОТА №3
УТИЛІТАРНІ КЛАСИ. ОБРОБКА МАСИВІВ І РЯДКІВ

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

1 ВИМОГИ

1.1 Розробник

- Котенко Сергій Миколайович;
- Студент групи КІТ 102.8(а);
- 15-11-2019р..

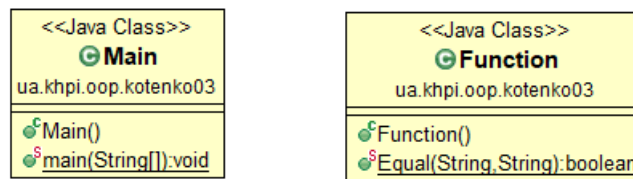
1.2 Загальне завдання

- 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) Продемонструвати 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).

Варіант 9. Ввести текст. Знайти та вивести, скільки разів повторюється в тексті кожне слово. Результат вивести у вигляді таблиці.

2 ОПИС ПРОГРАМИ

2.1 Опис логічної структури



2.2 Фрагменти коду

```
public static boolean Equal(String word_1,String word_2) {

    if(word_1.length() != word_2.length()) {
        return false;
    }

    for (int i = 0; i < word_1.length(); i++) {
        char ch_1 = word_1.charAt(i);
        char ch_2 = word_2.charAt(i);
        if(ch_1 != ch_2) {
            return false;
        }
    }
    return true;
}
```

Рисунок 2.1 – Порівняння слів

```
boolean exist = false;
Iterator<String> iter = un_words.iterator();
for(int i = 0; i<words.size();i++) {
    iter = un_words.iterator();
    exist = false;
    while(iter.hasNext()) {
        String value = iter.next();
        if(Function.Equal(words.get(i),value)) {
            exist = true;
            break;
        }
    }
    if(!exist) un_words.add(words.get(i));
}
```

Рисунок 2.2 – Знаходження унікальних слів

```

    int[] count_word = new int[un_words.size()];
    Iterator<String> it = un_words.iterator();
    for(int i = 0; i < un_words.size(); i++) {
        String word_1 = it.next();
        iter = words.iterator();
        while(iter.hasNext()) {
            String word_2 = iter.next();
            if(word_1.equals(word_2)) {
                count_word[i]++;
            }
        }
    }
}

```

Рисунок 2.3 – Підрахування кількості повторюваних слів

3 ВАРІАНТИ ВИКОРИСТАННЯ

3.1 Ілюстрація роботи програми

```

Enter your string:
Somebody once told me the world is gonna roll me
-----
| № |      Words      | Iteration |
-----
| 1 | Somebody        |    1    |
-----
| 2 | once            |    1    |
-----
| 3 | told            |    1    |
-----
| 4 | me              |    2    |
-----
| 5 | the             |    1    |
-----
| 6 | world           |    1    |
-----
| 7 | is              |    1    |
-----
| 8 | gonna           |    1    |
-----
| 9 | roll            |    1    |
-----

```

Рисунок 3.1 – Вивід результату

ВИСНОВОК

В інтегрованому середовищі *Eclipse* розроблена програма мовою Java. Виконання програми дозволяє продемонструвати роботу власних утилітарних класів та вирішення прикладних задач з використанням масивів і рядків.