

Лабораторна робота №3

Тема. Утилітарні класи. Обробка масивів і рядків

Мета. Розробка власних утилітарних класів, набуття навичок вирішення прикладних задач з використанням масивів і рядків.

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.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](#)).

1.1 Розробник

- Кононенко Дмитро Олексійович
- НТУ ХПИ КІТ 102.8а
- Варіант 8

1.2 Загальне завдання

- Ввести текст. У тексті знайти всі пари слів, з яких одне є обігом (словом навпаки) іншого (наприклад: “abc”-“cba”, “def”-“fed”). Результат вивести у вигляді таблиці.

2. Опис програми

- Програма розроблена для ознайомлення з IDE Eclipse та не має практичного застосування.

2.1 Засоби ООП

- Відсутні.

2.2 Ієрархія та структура класів

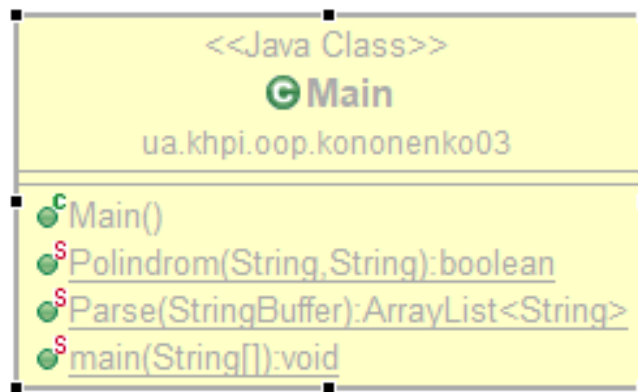


Рисунок 1.1 — діаграма класів.

2.3 Важливі фрагменти програми

```
public static ArrayList<String> Parse(StringBuffer parse_string){
    ArrayList<String> storage = new ArrayList<String>();
    String temp_storage;
    int temp = parse_string.indexOf(" ");
    while(parse_string.length() != 0) {

        temp_storage = new String(parse_string.substring(0,temp));
        temp_storage = temp_storage.trim();

        storage.add(temp_storage);

        parse_string = new StringBuffer(parse_string.substring(temp));

        temp = parse_string.indexOf(" ")+1;
        if(temp == 0) { // when the program trying to find last word

            temp = parse_string.length();

        }

    }

    return storage;
}
```

```

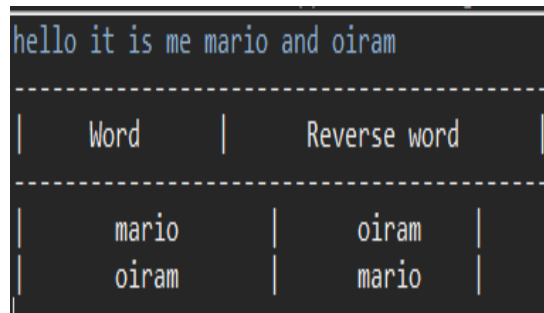
public static boolean Polindrom(String arg_1,String arg_2) {

    if(arg_1.length() != arg_2.length() | arg_1.equals("")) {
        return false;
    }

    for (int i = 0; i < arg_1.length(); i++) {
        char ch_1 = arg_1.charAt(i);
        char ch_2 = arg_2.charAt(arg_1.length() - 1 - i);
        if(ch_1 != ch_2) {
            return false;
        }
    }
    return true;
}

```

3.0 Варіанти використання



hello it is me mario and oiram	

Word	Reverse word

mario	oiram
oiram	mario

Рисунок 1.2 — результат роботи програми

3.1 Висновок

В інтегрованому середовищі *Eclipse* розроблена програма мовою Java. Виконання програми дозволяє виконувати пошук “паліндромів” .