НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Об’єктно-орієнтоване програмування»

Звіт з лабораторної роботи №3
Тема: «Утилітарні класи. Обробка масивів і рядків»

Виконав:
ст. гр. 1.КІТ102.8а
Міщенко Д.С.

Перевірив:
Пугачов Р.В.

Харків – 2019

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

1. Міщенко Дмитро Сергійович
2. 1.КІТ102.8а
3. Варіант 13

1.2 Загальне завдання

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).

Прикладні задачі

13. Ввести текст. Текст розбити на речення. Для кожного речення знайти та надрукувати всі слова максимальної та всі слова мінімальної довжини. Результат вивести у вигляді таблиці

2 ОПИС ПРОГРАМИ

2.1 Засоби ООП

Дана програма не використовує об’єктно-орієнтованих засобів.

2.2 Ієрархія та структура класів

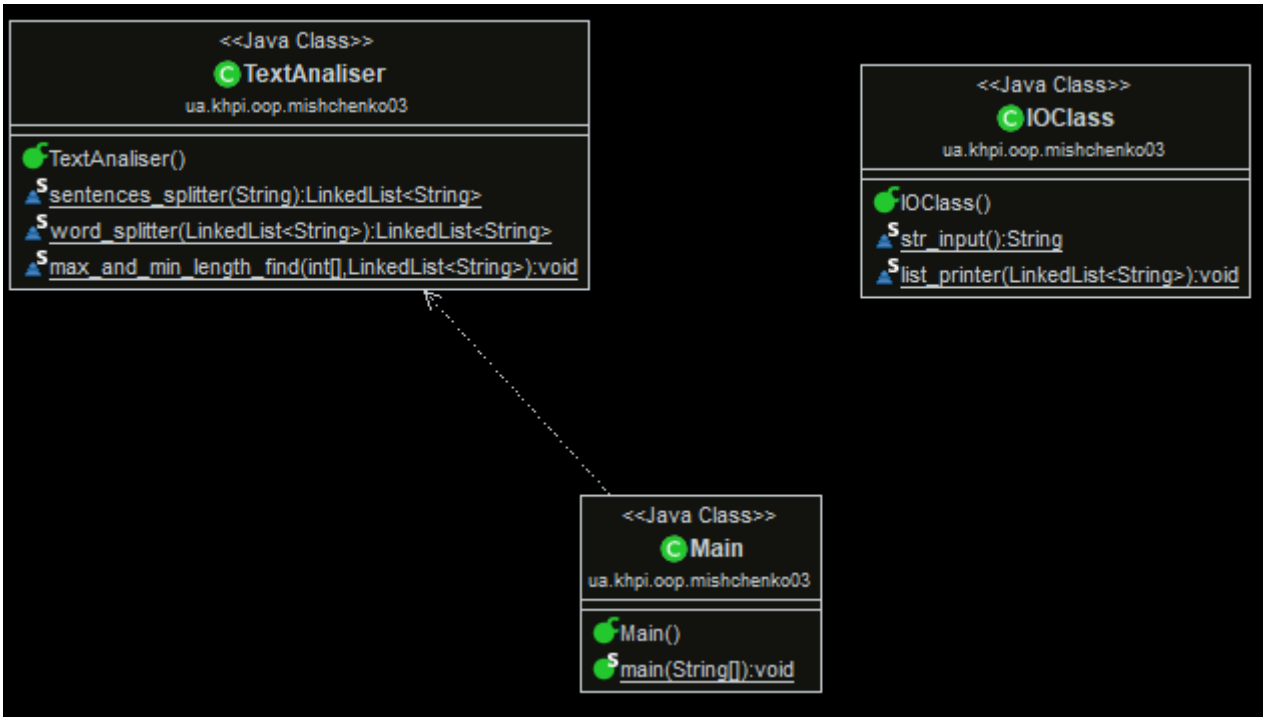

Рисунок 1 – Структура класів

2.3 Важливі фрагменти програми

```
static LinkedList<String> sentences_splitter(String text) {
    LinkedList<String> c = new LinkedList<String>();
    int begin = 0;
    for(int i = 0; i < text.length(); i++) {
        if(text.charAt(i) == '.' || text.charAt(i) == '?' || text.charAt(i) == '!') {
          if((text.length() - i) > 2) {
            if(text.substring(i,i+3).equals("..."))
            {
                String temp = text.substring(begin, i);
                temp = temp.trim();
                c.add(temp);
                begin = i+3;
                i+=3;
            }
            else
            {
            String temp = text.substring(begin,i);
            temp = temp.trim();
            c.add(temp);
            begin = i+1;
            }
          }else {
            String temp = text.substring(begin,i);
            temp = temp.trim();
            c.add(temp);
          }

        }
    }
    return c;
}
```

Рисунок 2 – Метод класу **TextAnaliser** *sentences_splitter*

```
static LinkedList<String> word_splitter(LinkedList<String> text) {
    LinkedList<String> words = new LinkedList<String>();
    Iterator<String> it = text.iterator();
    int[] words_in_sentence = new int[text.size()];
    int counter = 0;
    while(it.hasNext()) {

          String value = it.next();
          int begin = 0;
           for(int i = 0; i < value.length(); i++) {
              if(i > 0) {
              if((value.charAt(i) == ' ' || value.charAt(i) == ',' || value.charAt(i) == '-' || value.charAt(i) == ':
                  String temp = value.substring(begin,i);
                  temp = temp.trim();
                  words.add(temp);
                  begin = i+1;
                  words_in_sentence[counter]++;
              }

              if(i == value.length()-1) {
                  String temp = value.substring(begin,i+1);
                  temp = temp.trim();
                  words.add(temp);
                  words_in_sentence[counter]++;
                  break;
              }
              }

           }
           counter++;


      }
    max_and_min_length_find(words_in_sentence, words);
    return words;
  }
```

Рисунок 3 – Метод класу **TextAnaliser** *word_splitter*

3 ВАРІАНТИ ВИКОРИСТАННЯ

Програма знаходить в кожному реченні тексту слова найменшої та найбільшої довжини(рис.3).

```
String text = "Every story has a beginning and an end. "
        + "Each story has its own outline, synopsis, content, key points, prologues and epilogues. "
        + "And there is no such book in which with each new reading things, that had not been paid attention to before,"
        + "would not be revealed. "
        + "Every story has a beginning and an end. "
        + "Almost every ...";
```

Рисунок 4 – Приклад вхідних даних

```
-----------------------------------------
| Sentence |       Min length words      |
-----------------------------------------
|    1     |   a
|    2     |   has its own key and
|    3     |   is no in to be
|    4     |   a
|    5     |   every
-----------------------------------------
-----------------------------------------
| Sentence |       Max length words      |
-----------------------------------------
|    1     |   beginning
|    2     |   prologues epilogues
|    3     |   attention
|    4     |   beginning
|    5     |   Almost
-----------------------------------------
```

Рисунок 4 – Результат виконання програми

ВИСНОВКИ

В результаті виконання лабораторної роботи розробили власний утилітарний клас, набули навичок вирішення прикладних задач з використанням масивів і рядків.