

# 1. Тема роботи

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

### 1.1 Розробник

Гордон Андрій Андрійович

КН-108

Варіант 7

### 1.2 Загальне завдання

1. Розробити та продемонструвати консольну програму мовою *Java* 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”: 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.

5. Продемонструвати використання об'єктів класу `StringBuilder` або `StringBuffer`.
6. Застосувати функціональну (процедурну) декомпозицію – розробити власні утилітарні класи 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`).

### 1.3 Задача

Ввести текст. З тексту видалити всі слова заданої довжини, що починаються на приголосну літеру. Вивести початковий текст та результат.

## 2. ОПИС ПРОГРАМИ

### 2.1 Засоби ООП

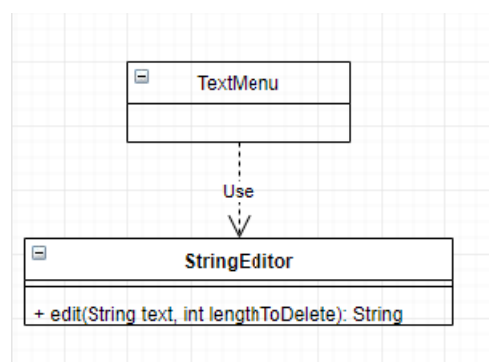
Декомпозиція для розділення завдання між класами.

### 2.2 Ієрархія та структура класів

Програма складається з двох класів.

`TextMenu` реалізує текстове меню користувача для введення даних та відображення результату.

`StringEditor` утилітарний клас для обробки даних згідно варіанту завдання.



### 2.3 Важливі фрагменти програми

## Обробка даних користувача

```
StringBuilder editedText = new StringBuilder();  
// відредагувати задану стрічку згідно з умовою  
for(int i = 0; i < text.length(); i++){  
    if(text.charAt(i) == ' '){  
        continue;  
    }  
  
    int counter = 0;  
    // підрахунок довжини слова  
    for(int j = i; j < text.length(); j++){  
        if(text.charAt(j) == ' '){  
            break;  
        }  
        counter++;  
    }  
  
    String word = text.substring(i, i + counter);  
    if(debug) System.out.println("Слово, що перевіряється: " + word);  
    // додати слово, якщо воно не задовільняє умову видалення  
    if(counter != deleteLength || f(word)){  
        editedText.append(word);  
        editedText.append(' ');  
    }  
    i += counter;  
    if(debug) System.out.println(editedText.toString());  
}
```

## ВИСНОВКИ

В ході лабораторної роботи, я розробив консольну програму на java для реалізації діалогового режиму роботи з користувачем. Вирішив прикладну задачу з використанням масивів та рядків.