

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА**  
**ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту



**Лабораторна робота N1**

З дисципліни

«Об'єктно-орієнтоване програмування»

**Виконав:**

Студент групи КН-108

Пулик Максим

**Викладач:**

Грабовська Н.Р.

Мета:

- 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. 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. Використовуючи програму рішення завдання відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню: а. введення даних; б. перегляд даних; с. виконання обчислень; d. відображення результату; е. завершення програми і т.д.

3. Забезпечити обробку параметрів командного рядка для визначення режиму роботи програми:

а. параметр “-h” чи “-help”: відображається інформація про автора програми, призначення (індивідуальне завдання), детальний опис режимів роботи (пунктів меню та параметрів командного рядка);

б. 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.

4. При вирішенні прикладних задач використовувати латинку .

5. Продемонструвати використання об’єктів класу StringBuilder або StringBuffer

6. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. Helper Class ) 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 ).

### 1.1 Розробник

Пулик Максим, КН-108, номер варіанту індивідуального завдання – 7.

### 1.2 Задача

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

## Опис програми:

Дана розроблена програма дозволяє обробляти(відповідно до індивідуального завдання) введений текст, реалізована робота двох можливих режимів роботи:

-h – режим при якому на початку програма з'являється певна інформація(інформація про розробника, суть індивідуального завдання, можливі функції консольного меню), -d або ж -debug – спеціальний режим роботи, при якому користувачу виводить на екран проміжний результат редагування рядка.

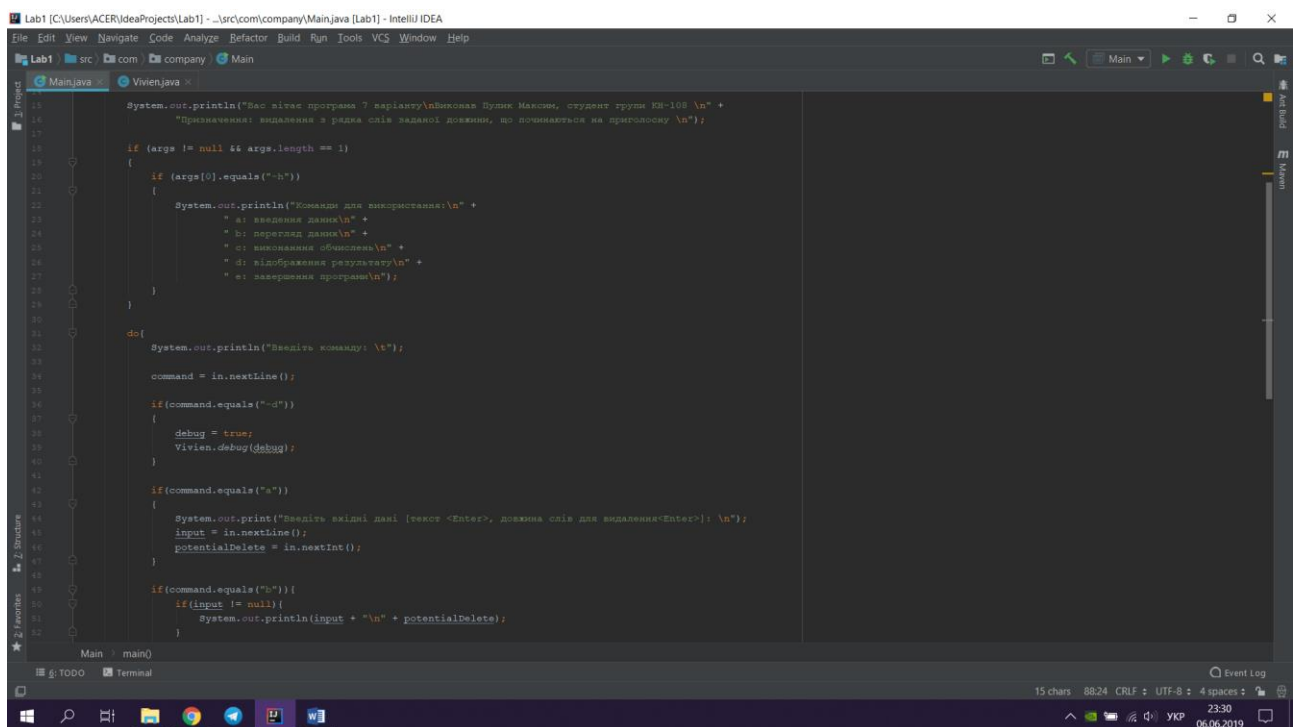
## 2.1 Засоби ООП

Для виконання завдання був використаний утилітарний клас Vivien, який містив певні статичні поля і методи необхідні для обрахунків.

## 2.2 Ієрархія та структура класів

Main – меню роботи з програмою.

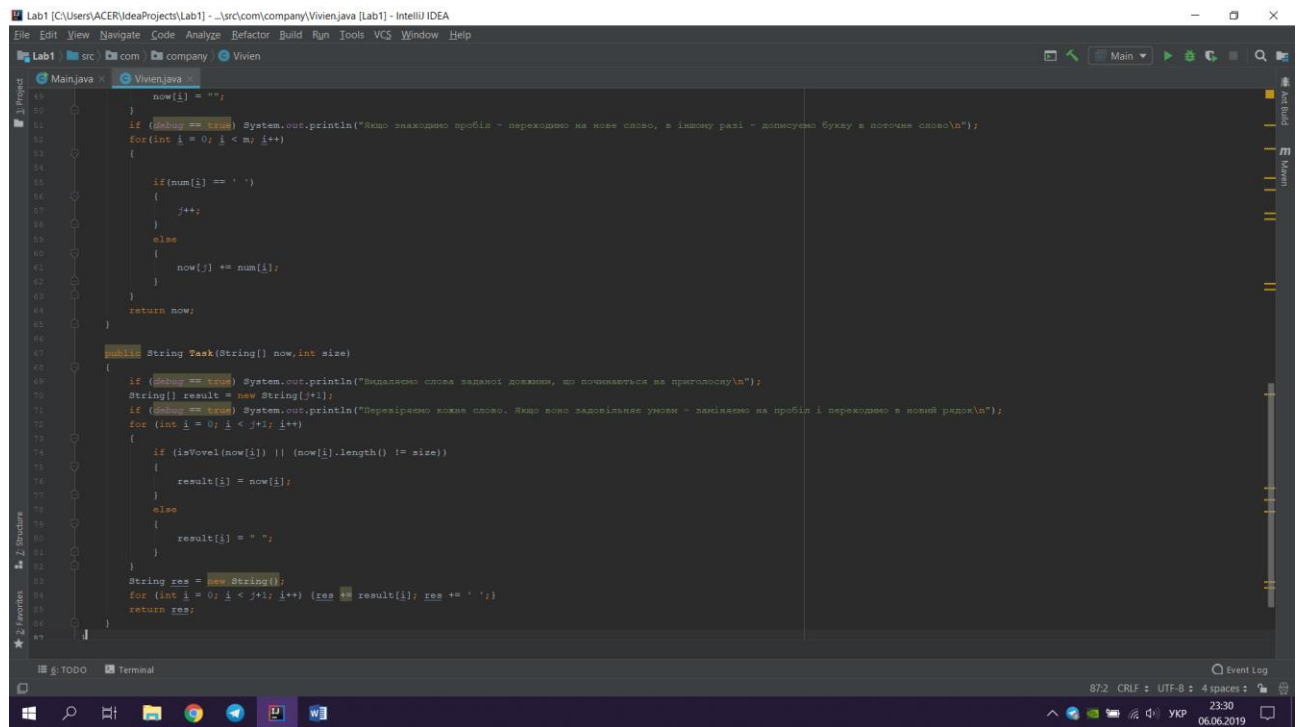
Vivien – клас з методами обробки тексту.



```
13 System.out.println("Бачиш програма ? варіанту\нВиконав Бучин Максим, студент групи КМ-108 \n" +
14 "Призначення: видалення з рядка слів заданої довжини, що починаються на приголосну \n");
15
16 if (args != null && args.length == 1)
17 {
18     if (args[0].equals("-h"))
19     {
20         System.out.println("Команди для використання:\n" +
21             " a: введення даних\n" +
22             " b: перетворення даних\n" +
23             " c: виконання обчислень\n" +
24             " d: відображення результату\n" +
25             " e: завершення програми\n");
26     }
27 }
28
29 do{
30     System.out.println("Введіть команду: \n");
31     command = in.nextLine();
32
33     if(command.equals("-d"))
34     {
35         debug = true;
36         Vivien.debug(debug);
37     }
38
39     if(command.equals("-e"))
40     {
41         System.out.print("Введіть зміни дані (текст <Enter>, довжина слів для видалення<Enter>): \n");
42         input = in.nextLine();
43         potentialDelete = in.nextInt();
44     }
45
46     if(command.equals("-b")){
47         if(input != null){
48             System.out.println(input + "\n" + potentialDelete);
49         }
50     }
51 }
```

```
Lab1 [C:\Users\ACER\IdeaProjects\Lab1] - \src\com\company\Main.java [Lab1] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Lab1 src com company Main
Main.java
13  if(input != null){
14      System.out.println(input + "\n" + potentialDelete);
15  }
16  else{
17      System.out.println("Введіть правильні дані!");
18  }
19  }
20  if(command.equals("c"))
21  {
22      if(input != null){
23          Vivien splin = new Vivien();
24          String[] sp = splin.Splinter(input);
25          output = splin.Task(sp,potentialDelete);
26          System.out.println("Дії над вхідними даними виконано. Введіть 'd' щоб побачити результати");
27      }
28      else{
29          System.out.println("Введіть правильні дані!");
30      }
31  }
32  if(command.equals("d"))
33  {
34      if(output != null)
35      {
36          System.out.println("Результат: ");
37          System.out.println(input);
38          System.out.println(output);
39          input = null;
40          output = null;
41      }
42      else
43      {
44          System.out.println("Введіть правильні дані!");
45      }
46  }
47  }while (!command.equals("e"));
48  System.exit(0);
Main main()
15 chars 88:24 CRLF UTF-8 4 spaces 23:30 06.06.2019
```

```
Lab1 [C:\Users\ACER\IdeaProjects\Lab1] - \src\com\company\Vivien.java [Lab1] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Lab1 src com company Vivien
Main.java Vivien.java
13  public boolean isVowel(String word)
14  {
15      if (debug == true) System.out.println("Перевіряємо наявність голосних літер на початку слова\n");
16      word = word.toLowerCase();
17      char c = word.charAt(0);
18      char [] chars = {'a','e','i','o','u','y'};
19      if (debug == true) System.out.println("Перевіряємо кожну першу букву слова\n");
20      if (debug == true) System.out.println("Якщо перша буква голосна - повернемо 'true'\n");
21      for (char a: chars)
22      {
23          if (a == c)
24          {
25              return true;
26          }
27      }
28      return false;
29  }
30  }
31  public String[] Splinter(String input){
32      if (debug == true) System.out.println("Розділяємо рядок на слова\n");
33      int m = input.length();
34      char[] num = input.toCharArray();
35      j=0;
36      int counter = 0;
37      for(int k = 0;k<input.length();k++)
38      {
39          if(input.charAt(k) == ' ')
40          {
41              counter++;
42          }
43      }
44      String[] now = new String[counter+1];
45      for(int i = 0;i<now.length;i++)
46      {
47          now[i] = "";
48      }
49      if (debug == true) System.out.println("Всіх знайдених слівів з розділювачем на кожен слів, в кожному сліві з розділювачем кожен з слівів слівів");
50  }
87:2 CRLF UTF-8 4 spaces 23:29 06.06.2019
```



### 3. Варіанти використання

Програма виконує певний набір заданих у завданні функцій.

## ВИСНОВКИ

У ході роботи розвинулись навички написання власних утилітарних класів та вирішення з їхньою допомогою задач над рядками з використанням класу `StringBuilder`. Також реалізовано можливість команд з боку користувача, а, крім цього, обробки командного рядка для визначення режиму роботи програми