

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**  
**Кафедра систем штучного інтелекту**



**Звіт до лабораторної 8**  
**з дисципліни**  
**“ ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ ”**

**Виконала:**  
**студентка групи КН-108**  
**інституту ІКНІ**  
**Бокшо Каріна**  
**Викладач:**  
**Гасько Р.Т.**

**Львів-2018**

Лабораторна робота №8  
Утилітарні класи Java SE. Обробка масивів і рядків. Інтерактивні  
консольні програми для платформи

## Мета 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. Індивідуальне завдання (1 варіант):

Ввести текст. Визначити та вивести, яких літер (голосних чи приголосних) більше в кожному реченні тексту. Результат вивести у вигляді таблиці.

### 1.1 Виконала:

Студентка Бокшо К. Е.; КН-108;

### 1.2 Рекомендації / вимоги до лабораторної роботи:

Вимоги

1. Розробити та продемонструвати консольну програму мовою *Java* в середовищі *Eclipse* для вирішення прикладної задачі за номером, що відповідає номеру студента в журналі групи з поверненням до початку. Наприклад **1 ->1, 2->2, ..., 15->15, 16->1, 17->2, ..., 30->15, 31-1** і т.д.
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;
  - д. 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. 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` )

## 2. Розробка програми

### 2.1 Ієрархія та структура класів

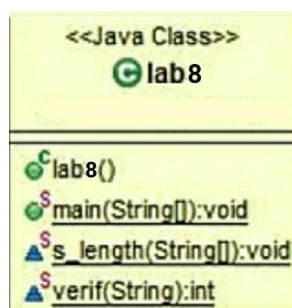
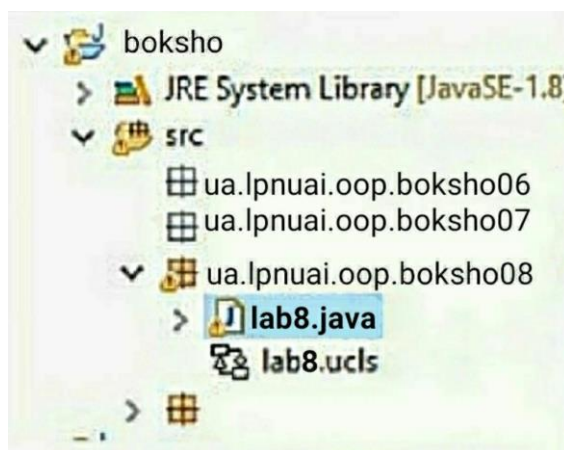


Рисунок 1 «Ієрархія та структура класів»

Структура розробленого проекту:



### 2.2 Опис програми

Програма реалізована у вигляді інтерактивного текстового меню з підтримкою команд користувача.

Основне призначення: використовуючи введений текст, визначає та виводить у вигляді таблиці, яких літер (голосних чи приголосних) більше в

кожному реченні тексту.

Для користувача доступні наступні команди:

- ENTRY - введення даних;
- VIEW - перегляд даних;
- CALC - виконання обчислень;
- RESULT - відображення результату;
- EXIT - завершення програми.

Програма працює лише з текстом написаним на латинкою. Для обробки даних використовуються класи-утиліти. Регулярних вирази не використовуються при виконанні завдання.

Також забезпечено обробку параметрів командного рядка для визначення режиму роботи програми:

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

Програма реалізована у вигляді інтерактивного текстового меню з підтримкою команд користувача.

Основне призначення: використовуючи введений текст, визначає та виводить у вигляді таблиці, яких літер (голосних чи приголосних) більше в кожному реченні тексту.

Для користувача доступні наступні команди:

- ENTRY - введення даних;
- VIEW - перегляд даних;
- CALC - виконання обчислень;
- RESULT - відображення результату;
- EXIT - завершення програми.

Програма працює лише з текстом написаним на латинкою. Для обробки даних використовуються класи-утиліти. Регулярних вирази не використовуються при виконанні завдання.

Також забезпечено обробку параметрів командного рядка для визначення режиму роботи програми:

## 2.3 Важливі фрагменти програми

```
public class Main {
    public static String LINE = "-----"
        + "-----\n";
    public static Scanner sc = new Scanner(System.in);
    public static boolean isDebug = false;
    public static String text = "";
    public static ArrayList<Integer> result = new ArrayList<Integer>();
    public static boolean exit = false;

    public static void main(String[] args) throws Exception {
        for (String comand: args) {
            if (comand.equals("-h") || comand.equals("-help")) {
                System.out.println(LINE);
                System.out.format("Автор: студент Бокшо К.Е.; КН-108; \n\n");
                System.out.format("Призначення програми: Визначення та"
                    + " виведення, яких літер (голосних чи приголосних)\n"
                    + "більше в кожному реченні введеного тексту.\n\n");
                System.out.format("Опис пунктів меню та параметрів командного"
                    + " рядка:\n\n");
                System.out.format("ENTRY - команда для вводу вхідних даних у"
                    + " вигляді тексту\n\n");
                System.out.format("VIEW - команда перегляд введених даних"
                    + "(відображає текст)\n\n");
                System.out.format("CALC - команда, що виконує пошук та"
                    + " підрахунок голосних та приголосних у кожному"
                    + " реченні тексту.\n\n");
                System.out.format("RESULT - відображення результату обчислень"
                    + " у вигляді таблиці\n\n");
                System.out.format("EXIT - завершення програми, вихід\n\n");
                System.out.format("-h або -help відображається інформація про"
                    + " автора програми, \n призначення (індивідуальне"
                    + " завдання), детальний опис режимів роботи\n "
                    + "(пунктів меню та параметрів командного"
                    + " рядка);\n\n");

                System.out.format("-d або -debug в процесі роботи програми"
                    + " відображаються додаткові дані, \n що полегшують"
                    + " налагодження та перевірку працездатності "
                    + "програми:\n діагностичні повідомлення, проміжні "
                    + "значення змінних, \n значення тимчасових змінних"
                    + " та ін.\n\n");
                System.out.println(LINE);
            }
            if (comand.equals("-d") || comand.equals("-debug")) {
                isDebug = true;
            }
        }

        System.out.format("Список доступних команд:\n\n");
        System.out.format("ENTRY - введення даних\n");
        System.out.format("VIEW - перегляд даних\n");
        System.out.format("CALC - виконання обчислень\n");
        System.out.format("RESULT - відображення результату\n");
        System.out.format("EXIT - завершення програми\n");
    }
}
```

```

    do {
        System.out.format("\n  Введіть команду: ");
        ACTION cur = ACTION.valueOf(sc.next());
        handleAction(cur);
    } while (!exit);
}

public static void handleAction(ACTION action) throws Exception {
    switch (action) {
        case ENTRY:
            if(Main.isDebugEnabled) {
                System.out.format("      Команда прийнята!\n");
            }
            System.out.format("\n  Введіть будь-ласка текст(латинкою):\n\n");
            BufferedReader br = new BufferedReader(
                new InputStreamReader(System.in));
            text = br.readLine(); // Збереження тексту з буферу
            if(Main.isDebugEnabled) {
                System.out.format("\nБуфер успішно створено!\n");
                System.out.format("\nТекст з буферу збережено!\n");
            }
            break;
        case VIEW:
            if(Main.isDebugEnabled) {
                System.out.format("      Команда прийнята!\n");
            }
            System.out.format("\n\n%s\n\n", text);
            if(Main.isDebugEnabled) {
                System.out.format("\nТекст успішно виведено!\n");
            }
            break;
        case CALC:
            if(Main.isDebugEnabled) {
                System.out.format("      Команда прийнята!\n");
            }
            result = TextHelper.handleText(text);
            break;
        case RESULT:
            if(Main.isDebugEnabled) {
                System.out.format("      Команда прийнята!\n");
            }
            ChartHelper.printChart(result);
            break;
        case EXIT:
            if(Main.isDebugEnabled) {
                System.out.format("      Команда прийнята!\n");
            }
            exit = true;
            break;
        default:
            System.err.println("Error data ^-(");
            break;
    }
}

public enum ACTION {
    ENTRY, VIEW, CALC, RESULT, EXIT
}

}

/**
 * class TextHelper - Утилітарний клас, що опрацьовує текст.
 *
 */
class TextHelper {
    public static String LINE = "-----"

```

```

        + "-----\n";
private static final char DOT = '.'; // Крапка
private static final char EXCLAMATION = '!'; // Знак оклику
private static final char QUESTION = '?'; // Знак питання
private static int sentences = 0;

/* Розбиває отриманий текст на речення */
public static ArrayList<Integer> handleText(String text) {

    /* Список, що зберігає результат */
    ArrayList<Integer> result = new ArrayList<Integer>();
    if(Main.isDebugEnabled) {
        System.out.format("\nСтворено список для"
            + " збереження результату!\n\n");
        System.out.format("Початок опрацювання тексту!\n\n");
    }
    String temp = ""; // Буфер
    for (int i = 0; i < text.length(); ++i) {
        char sign = text.charAt(i);
        if (sign == DOT || sign == EXCLAMATION || sign == QUESTION) {
            sentences++;
            if(Main.isDebugEnabled) {
                System.out.format("Знайдено речення №%d:"
                    + "\n%s\n\n", sentences, temp);
            }
            /* Кількість голосних */
            int vowels = StringHelper.countVowel(temp);
            /* Кількість приголосних */
            int consonants = StringHelper.countConsonants(temp);
            result.add(vowels);
            result.add(consonants);
            if(Main.isDebugEnabled) {
                System.out.format("Речення №%d знайдено:", sentences);
                System.out.format("    голосних - %d", vowels);
                System.out.format("    приголосних - %d\n", vowels);
                System.out.format("Результат збережено!\n");
                System.out.format("%s\n\n", LINE);
                if(i < text.length() - 1){
                    System.out.format("Опрацювання нового речення!\n\n");
                }
            }
            temp = "";
        } else {
            /*if(Main.isDebugEnabled) {
                System.out.format("До речення додано символ:"
                    + " %s\n\n", text.charAt(i));
            }*/
            temp += text.charAt(i);
        }
    }
    if(Main.isDebugEnabled) {
        System.out.format("Текст успішно опрацьовано!\n\n");
    }
    return result;
}

/**
 * class StringHelper - Утилітарний клас, що виконує пошук та
 * підрахунок голосних і приголосних у реченні.
 *
 */
class StringHelper {
    /* Перелік голосних */
    private static final String VOWELS = "aeiouyAEIOUY";
    /* Перелік приголосних */
    private static final String CONSONANTS = "bcdfghjklmnpqrstvwxyz"

```

```

        + "BCDFGHJKLMNPQRSTVWXZ";

    /* Перевіряє чи є символ голосною буквою */
    public static boolean isVowel(char ch) {

        return VOWELS.indexOf(ch) >= 0;
    }

    /* Перевіряє чи є символ приголосною буквою */
    public static boolean isConsonants(char ch) {

        return CONSONANTS.indexOf(ch) >= 0;
    }

    /* Підраховує голосні */
    public static int countVowel(String sentence) {

        if(Main.isDebugEnabled) {
            System.out.format("Розпочато пошук і підрахунок голосних!\n\n");
        }
        int counter = 0;
        for (int i = 0; i < sentence.length(); ++i) {
            if (isVowel(sentence.charAt(i))) {
                if(Main.isDebugEnabled) {
                    System.out.format("Знайдено голосну:"
                        + " %s\n\n", sentence.charAt(i));
                }
                ++counter;
            }
        }
        if(Main.isDebugEnabled) {
            System.out.format("Пошук і підрахунок голосних успішно"
                + " закінчено!\n\n");
        }
        return counter;
    }

    /* Підраховує приголосні */
    public static int countConsonants(String sentence) {

        if(Main.isDebugEnabled) {
            System.out.format("Розпочато пошук і підрахунок приголосних!\n\n");
        }
        int counter = 0;
        for (int i = 0; i < sentence.length(); ++i) {
            if (isConsonants(sentence.charAt(i))) {
                if(Main.isDebugEnabled) {
                    System.out.format("Знайдено приголосну:"
                        + " %s\n\n", sentence.charAt(i));
                }
                ++counter;
            }
        }
        if(Main.isDebugEnabled) {
            System.out.format("Пошук і підрахунок приголосних успішно"
                + " закінчено!\n\n");
        }
        return counter;
    }
}

/**
 * class ChartHelper - Утилітарний клас, що виконує виведення результатів.
 *
 */
class ChartHelper {
    public static String LINE = "-----"

```



```

+ "-----\n";

/* Виводить дані у вигляді таблиці */
public static void printChart(ArrayList<Integer> data) {
    int counter = 0;
    System.out.println(LINE);
    if(Main.isDebugEnabled) {
        System.out.format("Розпочато виведення даних!\n\n");
    }
    System.out.format("      Реченя №      Голосних      Приголосних\n\n");
    for (int i = 0; i < data.size(); i += 2) {
        counter++;
        System.out.format("      %d      %d      %d\n", counter,
data.get(i),
                        data.get(i + 1));
        if(Main.isDebugEnabled) {
            System.out.format("Дані списку під індексом [%d] та [%d]"
+ " виведено!\n\n", i, i + 1);
        }
    }
    if(Main.isDebugEnabled) {
        System.out.format("Виведення даних успішно закінчено!\n\n");
    }
    System.out.format("\n%s", LINE);
}
}

```

### 3. РЕЗУЛЬТАТ РОБОТИ

Для налагодження роботи програми було успішно проведено її тестування.

```
Список доступних команд:

ENTRY - введення даних
VIEW - перегляд даних
CALC - виконання обчислень
RESULT - відображення результату
EXIT - завершення програми

Введіть команду: ENTRY

Введіть будь-ласка текст(латинкою):

There are many big and small libraries everywhere in our country. They have millions of books in different languages.
You can find there the oldest and the newest books. Every school has a library. Pupils come to the library to take books on different subjects. The school library where Oleg studies is good. It is a large clean room. There are four big windows in it. The walls are light blue. There are a lot of shelves full of books. You can find books on literature, physics, history, chemistry, geography, biology and other subjects. There are books in English, too. On the walls you can see pictures of some great writers and poets. On the table near the window you can always see beautiful spring and autumn flowers.

Введіть команду: CALC

Введіть команду: RESULT
-----
Речення №      Голосних      Приголосних
1               24             30
2               18             25
3               16             25
4               10             12
5               20             32
6               16             22
7               9              10
8               11             15
9               7              13
10              13             19
11              34             48
12              11             14
13              22             31
14              30             39
-----

Введіть команду: _
```

Рисунок 2 "Результат роботи

```
C:\Windows\system32\cmd.exe

Текст успішно опрацьовано!

Введіть команду: RESULT
Команда прийнята!
-----
Розпочато виведення даних!

Речення №      Голосних      Приголосних
1               24             30
Дані списку під індексом [0] та [1] виведено!
2               18             25
Дані списку під індексом [2] та [3] виведено!
3               16             25
Дані списку під індексом [4] та [5] виведено!
4               10             12
Дані списку під індексом [6] та [7] виведено!
5               20             32
Дані списку під індексом [8] та [9] виведено!
6               16             22
Дані списку під індексом [10] та [11] виведено!
7               9              10
Дані списку під індексом [12] та [13] виведено!
8               11             15
Дані списку під індексом [14] та [15] виведено!
9               7              13
Дані списку під індексом [16] та [17] виведено!
10              13             19
Дані списку під індексом [18] та [19] виведено!
11              34             48
Дані списку під індексом [20] та [21] виведено!
12              11             14
Дані списку під індексом [22] та [23] виведено!
13              22             31
Дані списку під індексом [24] та [25] виведено!
14              30             39
Дані списку під індексом [26] та [27] виведено!
Виведення даних успішно закінчено!
-----

Введіть команду:
```

програми"

Створено і налагоджено програму, що повністю виконую поставлене

## **ВИСНОВКИ**

індивідуальне завдання та відповідає вимогам.

Було отримано і вдосконалено навички у розробці інтерактивних консольних програм мовою Java.