Лабораторна робота №3.

УТИЛІТАРНІ КЛАСИ. ОБРОБКА МАСИВІВ І РЯДКІВ

**Мета роботи:**

Розробка власних утилітарних класів. Набуття навичок вирішення прикладних задач з використанням масивів і рядків.

**1. Індивідуальне завдання:**

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

**1.1 Розробник:**

*студент Литвин Ігнатій Ігоревич; КІТ-26А; Варіант №7*

**1.2 Рекомендації / вимоги до лабораторної роботи:**

* 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 (Class String, Manipulating Characters in a String, Comparing Strings and Portions of Strings).
* При вирішенні прикладних задач використовувати латинку.
* Продемонструвати використання об'єктів класу StringBuilder або StringBuffer.
* Для обробки даних використовувати класи-утиліти (особливий випадок допоміжного класу, див. Helper Class).
* Забороняється використовувати засоби обробки регулярних виразів.

**2. Розробка програми**

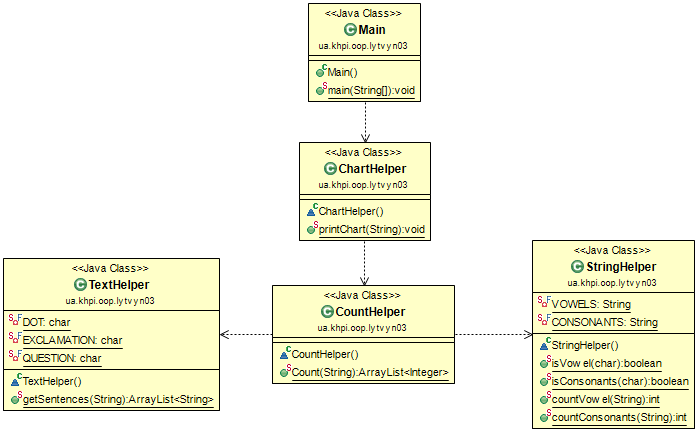
**2.1 Ієрархія та структура класів**

Рисунок 1 « Ієрархія та структура класів»

**2.2 Опис програми**

Програма реалізована у вигляді консольного вікна з послідовним виконанням завдання.

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

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

**2.3 Важливі фрагменти програми**

/\*\*

\* class TextHelper Утилітарний клас, що опрацьовує текст (розбиває його на

\* речення).

\*

\* **@author** student Lytvyn I.I. KIT-26A

\*

\*/

**class** TextHelper {

**private** **static** **final** **char** ***DOT*** = '.'; // Крапка

**private** **static** **final** **char** ***EXCLAMATION*** = '!'; // Знак оклику

**private** **static** **final** **char** ***QUESTION*** = '?'; // Знак питання

/\* Розбиває отриманний текст на речення \*/

**public** **static** ArrayList<String> getSentences(String text) {

/\* Список, що зберігає результат \*/

ArrayList<String> result = **new** ArrayList<String>();

String temp = ""; // Буфер

**for** (**int** i = 0; i < text.length(); ++i) {

**char** sign = text.charAt(i);

**if** (sign == ***DOT*** || sign == ***EXCLAMATION*** || sign == ***QUESTION***) {

result.add(temp);

temp = "";

} **else** {

temp += text.charAt(i);

}

}

**return** result;

}

}

/\*\*

\* class StringHelper Утилітарний клас, що виконує пошук та підрахунок голосних

\* та приголосних у речені.

\*

\* **@author** student Lytvyn I.I. KIT-26A

\*

\*/

**class** StringHelper {

/\* Перелік голосних \*/

**private** **static** **final** String ***VOWELS*** = "aeiouyAEIOUY";

/\* Перелік приголосних \*/

**private** **static** **final** String ***CONSONANTS*** = "bcdfghjklmnpqrstvwxz" + "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) {

**int** counter = 0;

**for** (**int** i = 0; i < sentence.length(); ++i) {

**if** (*isVowel*(sentence.charAt(i))) {

++counter;

}

}

**return** counter;

}

/\* Підраховує приголосні \*/

**public** **static** **int** countConsonants(String sentence) {

**int** counter = 0;

**for** (**int** i = 0; i < sentence.length(); ++i) {

**if** (*isConsonants*(sentence.charAt(i))) {

++counter;

}

}

**return** counter;

}

}

/\*\*

\* class CountHelper Утилітарний клас, що заповнює список данними для подальшого

\* опрацювання.

\*

\* **@author** student Lytvyn I.I. KIT-26A

\*

\*/

**class** CountHelper {

/\* Заносить кількість голосних та приголосних до списку \*/

**public** **static** ArrayList<Integer> Count(String text) {

ArrayList<Integer> result = **new** ArrayList<Integer>();

ArrayList<String> sentences = TextHelper.*getSentences*(text);

**for** (**int** i = 0; i < sentences.size(); i++) {

/\* Кількість голосних \*/

**int** vowels = StringHelper.*countVowel*(sentences.get(i));

/\* Кількість приголосних \*/

**int** consonants = StringHelper.*countConsonants*(sentences.get(i));

result.add(vowels);

result.add(consonants);

}

**return** result;

}

}

/\*\*

\* class ChartHelper Утилітарний клас, що виконує виведення результатів.

\*

\* **@author** student Lytvyn I.I. KIT-26A

\*

\*/

**class** ChartHelper {

/\* Виводить дані у вигляді таблиці \*/

**public** **static** **void** printChart(String text) {

ArrayList<Integer> data = CountHelper.*Count*(text);

**int** counter = 0;

System.***out***.println("---------------------------------------------"

+ "---------------------------\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));

}

System.***out***.println("\n--------------------------------------------"

+ "---------------------------\n");

}

**3. РЕЗУЛЬТАТ РОБОТИ**

Для налагодження роботи програми було успішно проведено її тестування.

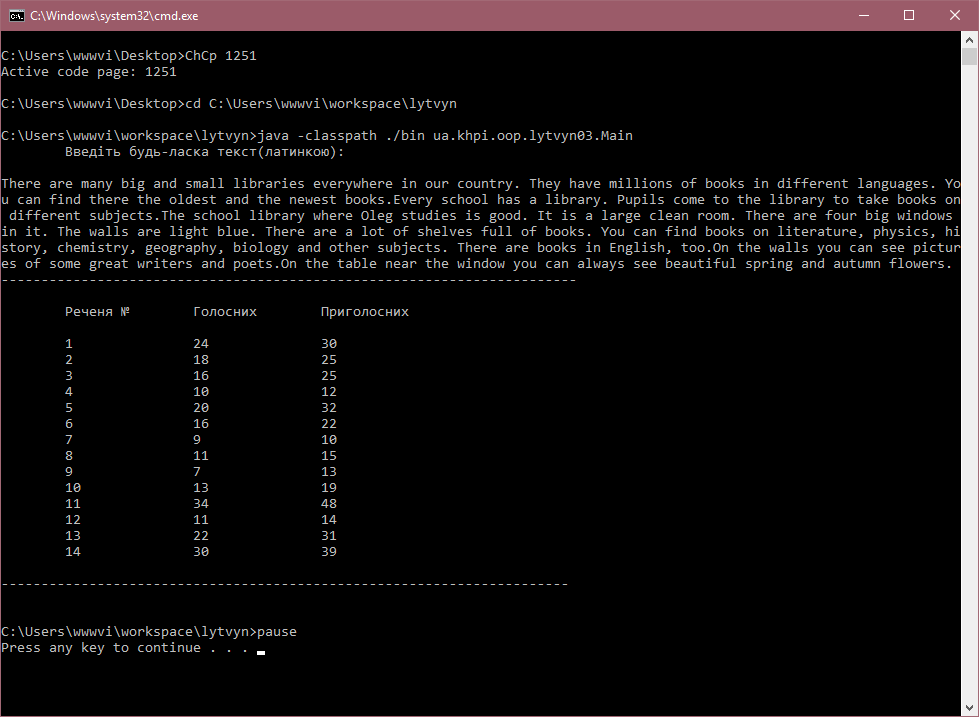


Рисунок 2 "Результат роботи програми"

**ВИСНОВКИ**

*Створено і налагоджено програму, що повністю виконую поставлене індивідуальне завдання та відповідає вимогам.   
Було отримано і вдосконалено навички у розробці власних утилітарних класів та у вирішенні прикладних задач з використанням масивів і рядків*