Лабораторна робота №4.

ІНТЕРАКТИВНІ КОНСОЛЬНІ ПРОГРАМИ ДЛЯ ПЛАТФОРМИ JAVA SE

**Мета роботи:**

Розробка інтерактивних консольних програм мовою Java. Реалізація діалогового режиму роботи з користувачем в консольних програмах.

**1. Завдання / вимоги до лабораторної роботи:**

1. Використовуючи програму рішення завдання лабораторної роботи №3, відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню:

* введення даних;
* перегляд даних;
* виконання обчислень;
* відображення результату;
* завершення програми і т.д.

1. Забезпечити 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.

**1.1 Розробник:**

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

**2. Розробка програми**

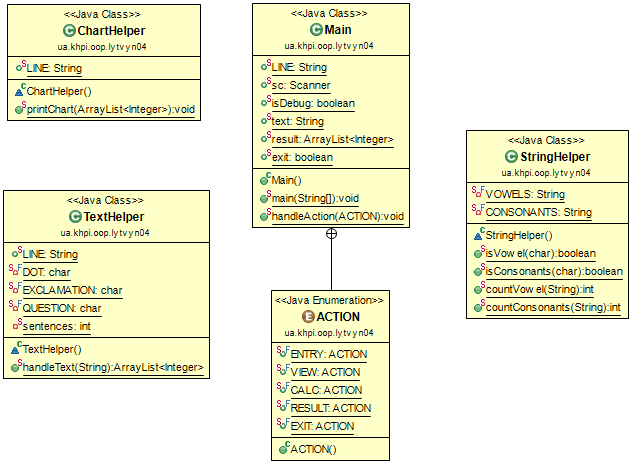
**2.1 Ієрархія та структура класів**

Рисунок 1 « Ієрархія та структура класів»

**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.

**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 КІТ-26А Литвин І.І.\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\n та приголосних у кожному"

+ " речені 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.*isDebug*){

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.*isDebug*){

System.***out***.format("\nБуфер успішно створено!\n");

System.***out***.format("\nТекст з буферу збережено!\n");

}

**break**;

**case** ***VIEW***:

**if**(Main.*isDebug*){

System.***out***.format(" Команда прийнята!\n");

}

System.***out***.format("\n\n%s\n\n", *text*);

**if**(Main.*isDebug*){

System.***out***.format("\nТекст успіщно виведено!\n");

}

**break**;

**case** ***CALC***:

**if**(Main.*isDebug*){

System.***out***.format(" Команда прийнята!\n");

}

*result* = TextHelper.*handleText*(*text*);

**break**;

**case** ***RESULT***:

**if**(Main.*isDebug*){

System.***out***.format(" Команда прийнята!\n");

}

ChartHelper.*printChart*(*result*);

**break**;

**case** ***EXIT***:

**if**(Main.*isDebug*){

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.

\*

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

\*

\*/

**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 отриманний текст на речення \*/

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

/\* Список, що зберігає результат \*/

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

**if**(Main.*isDebug*){

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.*isDebug*){

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.*isDebug*){

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.isDebug){

System.out.format("До речення додано символ:"

+ " %s\n\n", text.charAt(i));

}\*/

temp += text.charAt(i);

}

}

**if**(Main.*isDebug*){

System.***out***.format("Текст успішно опрацьовано!\n\n");

}

**return** result;

}

}

/\*\*

\* class StringHelper - Утилітарний клас, що виконує пошук та

\* підрахунок голосних і 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) {

**if**(Main.*isDebug*){

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.*isDebug*){

System.***out***.format("Знайдено голосну:"

+ " %s\n\n", sentence.charAt(i));

}

++counter;

}

}

**if**(Main.*isDebug*){

System.***out***.format("Пошук і підрахунок голосних успішно"

+ " закінчено!\n\n");

}

**return** counter;

}

/\* Підраховує приголосні \*/

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

**if**(Main.*isDebug*){

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.*isDebug*){

System.***out***.format("Знайдено приголосну:"

+ " %s\n\n", sentence.charAt(i));

}

++counter;

}

}

**if**(Main.*isDebug*){

System.***out***.format("Пошук і підрахунок приголосних успішно"

+ " закінчено!\n\n");

}

**return** counter;

}

}

/\*\*

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

\*

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

\*

\*/

**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.*isDebug*){

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.*isDebug*){

System.***out***.format("Дані списку під індексом [%d] та [%d]"

+ " виведено!\n\n", i, i + 1);

}

}

**if**(Main.*isDebug*){

System.***out***.format("Виведення даних успішно закінчено!\n\n");

}

System.***out***.format("\n%s",*LINE*);

}

}

1. **РЕЗУЛЬТАТ РОБОТИ**

Для налагодження роботи програми було успішно проведено її тестування.

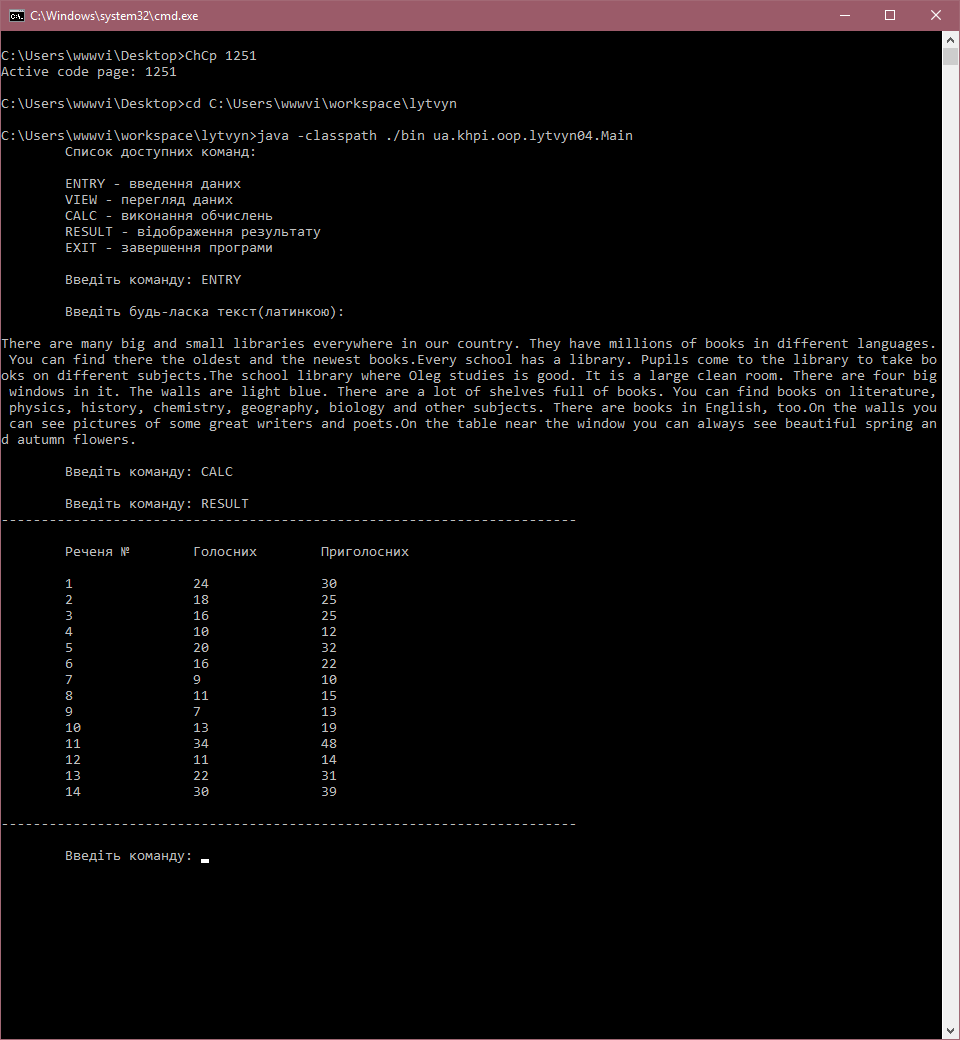


Рисунок 2 "Результат роботи програми"

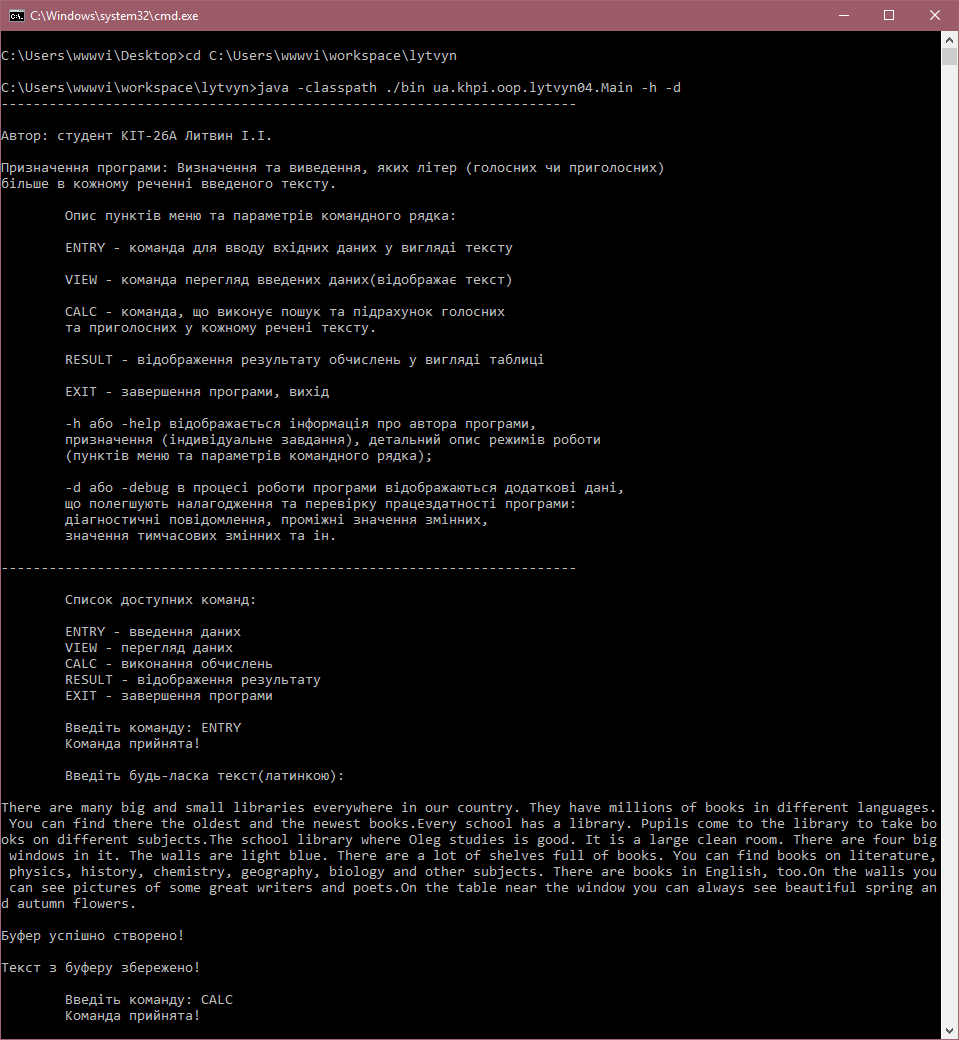
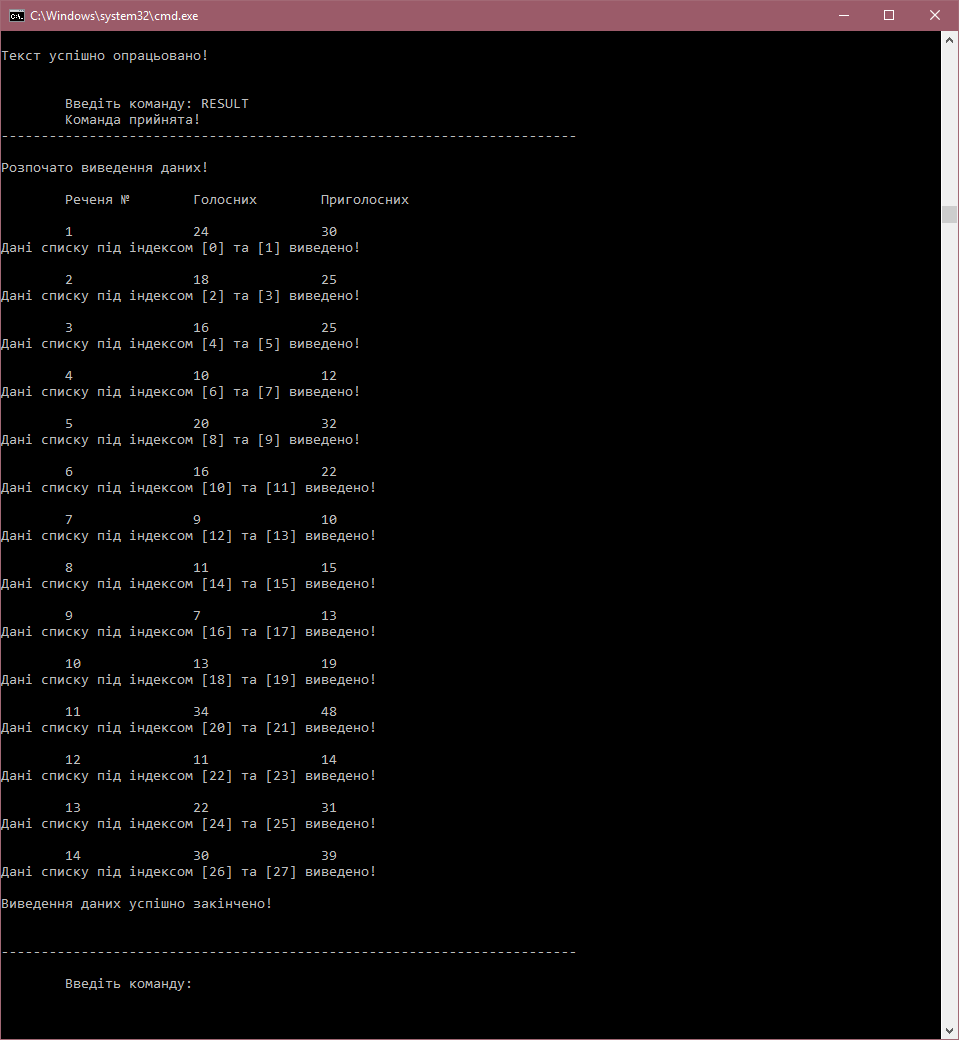
****

Рисунок 3 "Робота програми з параметрами -h -d"

Рисунок 4 "Робота програми з параметром -d"

****

**ВИСНОВКИ**

*Створено і налагоджено програму, що повністю виконую поставлене індивідуальне завдання та відповідає вимогам.   
Було отримано і вдосконалено навички у розробці інтерактивних консольних програм мовою Java.*