**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп’ютерних наук**

**Кафедра Інженерії програмного забезпечення**

**ЗВІТ**

*з лабораторної роботи № 1*

**“Керуючі конструкції, масиви, робота з класом String, використання систем контролю версій”**

Дисципліна "Об'єктно-орієнтоване програмування"

Спеціальність: **Інженерія програмного забезпечення**

121-ЛР.ПЗ.01-209.1910907

***Cтудент***

\_\_\_\_\_\_*В.М.Гільщанський*

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

***Ст. викл. кафедри ІПЗ***

*\_\_\_\_\_\_\_ \_\_Ю.О.Нездолій*

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

**Миколаїв – 2021**

**Лабораторна робота 1**

**Тема:** Керуючі конструкції, масиви, робота з класом String, використання систем контролю версій.

**Мета:** - ознайомити слухачів з основами використання Git при роботі у команді, з типовими ситуаціями при роботі та стандартними рішеннями

- дати можливість отримати досвід використання масивів та керуючих конструкцій для розв’язання практичних завдань

- ознайомити слухачів з основними можливостями класу String.

**Варіант 8. Набір завдань – 6, 8, 18, 23, 24.**

**Завдання 6:** Створіть метод, який дозволяє порівняти 2 будь яких масиви int[] по складу. Масиви можуть відрізнятися по порядку елементів , но бути однаковими зі складу. Приклад- [1,4,7]==[4,7,1] – еквівалентні.

**Лістинг коду:**

package com.company;  
import java.util.Arrays; //import the Arrays class  
  
public class Main {  
  
 //Завдання 6  
 //Створіть метод, який дозволяє порівняти 2 будь яких масиви int[] по складу.  
 //Масиви можуть відрізнятися по порядку елементів , но бути однаковими зі складу.  
 //Приклад - [1,4,7]==[4,7,1- еквівалентні.]  
 public static void arrContentCompare(int[] arr1,int[] arr2) {  
 int[] tempArr1=arr1; //Створення змінних - тимчасових копій масивів  
 int[] tempArr2=arr2;  
 Arrays.*sort*(tempArr1); //Сортування масивів  
 Arrays.*sort*(tempArr2);  
 if(Arrays.*equals*(tempArr1,tempArr2)) //Перевірка на вміст  
 {  
 System.*out*.println("The both arrays are the same"); //Якщо однаковий вміст  
 }  
 else{  
 System.*out*.println("The both arrays are different"); //В іншому випадку  
 }  
 }  
  
 public static void main(String[]args) {  
 //8 Варіант. Завдання 6,8,18,23,24.  
 //Завдання 6  
 int[]arr1 = {3,4,7};  
 int[]arr2 = {4,7,1};  
 *arrContentCompare*(arr1,arr2);  
 }  
}

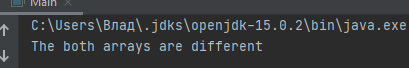
****

Рисунок 1 – Результат відпрацювання програми.

**Завдання 8:** Створіть метод, який підраховує кількість однакових чисел у будь- якому масиві int[]. Приклад [1,7,3,9,34,3,7] - результат : 1 - 1раз, 7 - 2 рази, 3 - 2 рази....

**Лістинг коду:**

package com.company;  
import java.util.Arrays; //import the Arrays class  
import java.util.HashMap; // import the HashMap class  
  
public class Main {  
 //Завдання 8  
 //Створіть метод, який підраховує кількість однакових чисел у будь якому масиві int[].  
 //Приклад [1,7,3,9,34,3,7] Результат : 1- 1раз, 7 - 2 рази,3-2 рази.  
 public static int digitCount(int[] arr,int num){ //Метод який буде підраховувати кількість чисел  
 int count = 0;  
 for (int i=0;i<arr.length;i++){  
 if (num==arr[i]){  
 count++;  
 }  
 }  
 return count;  
 }  
 public static void countSameNumbers(int[] arr){  
 HashMap<Integer,Integer> numbers = new HashMap<Integer,Integer>(); //Потрібен для зручного виведення результату  
 for (int i = 0;i < arr.length; i++){  
 numbers.put(arr[i], *digitCount*(arr, arr[i]));  
 }  
 System.*out*.println(numbers);  
 }  
  
 public static void main(String[]args) {  
 //8 Варіант. Завдання 6,8,18,23,24.  
 //Завдання 8  
 int[]arr3 = {1,7,3,9,34,3,7};  
 *countSameNumbers*(arr3);  
 }  
}



Рисунок 2 – Результат відпрацювання програми.

**Завдання 18:** Створіть метод, який приймає два параметри типу String та повертає кількість повних повторів одного слова літерами другого. Приклад - слово «він» можливо набрати три рази літерами, які є у рядку «ннніііввввукег».

**Лістинг коду:**

package com.company;  
import java.util.Arrays; //import the Arrays class  
import java.util.HashMap; // import the HashMap class  
  
public class Main {  
 //Завдання 18  
 //Створіть метод, який приймає два параметри типу String  
 //та повертає кількість повних повторів одного слова літерами другого.  
 //Приклад - слово =він= можливо набрати три рази літерами , які є у рядку =ннніііввввукег=  
 public static void countRepeat(String word,String someStr){  
 int repeats = 0; //Змінна яка буде зберігати кількість кожної літери яка є в рядку  
 int[]arrRepeats = new int[word.length()]; //Масив який буде зберігати кількість кожної літери яка є в рядку  
 int minRepeats; //Змінна яка буде зберігати мінімальну кількість повторювань літер  
 for (int j=0;j<word.length();j++) {  
 for (int i = 0; i < someStr.length(); i++) {  
 if (word.charAt(j) == someStr.charAt(i)) { //Первірка чи є відповідна літера в рядку  
 repeats++;  
 arrRepeats[j] = repeats;  
 }  
 }  
 repeats=0; //Обнулюємо "счетчик"  
 }  
 minRepeats = arrRepeats[0];  
 for (int i=1;i<arrRepeats.length;i++){ //Мета цього цилку, "обійти" кожний елемент масиву  
 if(minRepeats>arrRepeats[i]){ //І знайти мінімальне  
 minRepeats=arrRepeats[i];  
 }  
 }  
 System.*out*.println("Cлово - "+word+" - можливо набрати - "+minRepeats+" раз(и) літерами, які є у рядку: "+someStr);  
 }  
 public static void main(String[]args) {  
 //8 Варіант. Завдання 6,8,18,23,24.  
 //Завдання 18  
 String someStr="ннніііввввукег";  
 String word = "він";  
 *countRepeat*(word,someStr);  
 }  
}

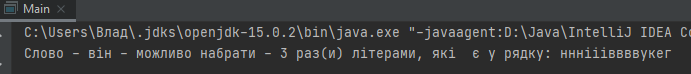
****

Рисунок 3 – Результат відпрацювання програми.

**Завдання 23:** Створіть метод, який рекурсивно рахує суму елементів масиву int[], який передається як параметр методу.

**Лістинг коду:**

package com.company;  
import java.util.Arrays; //import the Arrays class  
import java.util.HashMap; // import the HashMap class  
  
public class Main {  
 //Завдання 23  
 //Створіть метод, який рекурсивно рахує суму елементів масиву int[] , який передається як параметр методу  
 public static int recurseSum(int[] arr,int index){  
 if(index>=arr.length){ //Умова виходу з рекурсії  
 return 0;  
 }  
 return arr[index]+*recurseSum*(arr,index+1);  
 }  
 public static void main(String[]args) {  
 //Завдання 23  
 int[] arr4 = {13, 73, 25, 25, 100, 35};  
 System.*out*.println(*recurseSum*(arr4, 0));  
 }  
}

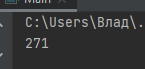
****

Рисунок 4 – Результат відпрацювання програми.

**Завдання 23:** Створіть метод, який сортує масив String у порядку довжини рядків.

**Лістинг коду:**

package com.company;  
import java.util.Arrays; //import the Arrays class  
import java.util.HashMap; // import the HashMap class  
  
public class Main {  
 //Завдання 24  
 //Створіть метод, який сортує масив String у порядку довжини рядків  
 public static void sortLengthStr(String[] str){  
 int len=str.length; //Змінна яка буде зберігати довжину рядка  
 for(int i = 0;i < len; i++){ //Два цикла, з умновною конструкцією яка перевіряє  
 for (int j = 0; j < len; j++){ //Довжинку кожного слова рядка  
 if(str[i].length()<str[j].length()){  
 String temp = str[i];  
 str[i]=str[j];  
 str[j]=temp;  
 }  
 }  
 }  
 for (int j = 0; j < len; j++){ //Виведення  
 System.*out*.println(str[j]);  
 }  
  
 }  
 public static void main(String[]args) {  
 //Завдання 24  
 String[] someWords = {"World","Was","Car","Knight","Moon","End"};  
 *sortLengthStr*(someWords);  
 }  
}

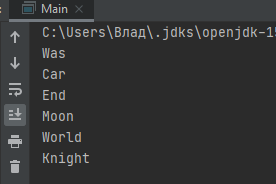
****

Рисунок 5 – Результат відпрацювання програми.

**Висновок:** в ході виконання лабораторної роботи були виконанні запропоновані завдання, отримали досвід у використанні масивів та керуючих конструкцій для розв’язання поставлених завдань. Ознайомилися з основними можливостями класу String. Також було застосовано деякі методи, та функції класу String та Array. Відправили звіт роботи у репозиторій GitHub.