**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

Учреждение образования

«Гомельский государственный технический университет

имени П.О.Сухого»

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

по дисциплине «ООП»

на тему: «Базовые навыки»

Выполнил: студент группы ИП-22

Коваленко А.И.

Проверил: ст. преподаватель

Шибеко В. Н

Гомель 2022

***Цель работы****: Получить навыки работы в среде разработки Java, научиться создавать классы, содержащие поля, методы, конструкторы, научиться создавать и использовать объекты класса.*

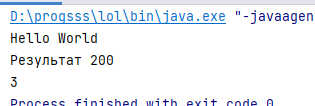
**Задание 1.** Знакомство со средами разработки

* Создать проект HelloWorld, получить результат вывода строки - System. out.println("Hello World");
* Добавить код, получить результат:
* int num=100;
* num=num\*2;
* system.out.println("Результат "+num);
* int arr2[][] = {{1,3},{2,4},{3,5}};
* int x = arr2[0][1];
* system.out.print(x)

Листинг main:

package com.company;  
  
public class main {  
  
 public static void main(String[] args)  
 {  
 System.*out*.println("Hello World");  
 int num = 100;  
 num\*= 2;  
 System.*out*.println("Результат "+ num);  
 int arr2[][] = {{1,3},{2,4},{3,5}};  
 int x = arr2[0][1];  
 System.*out*.print(x);  
  
 }  
  
}

Результат:



**Задание 2.** Базовые конструкции объектно-ориентированного программирования, массивы

Создать класс «LR\_1» с методами:

* создания массива
* метод, возвращающий длину массива,
* метод, сортирующий массив,
* метод для доступа к элементу массива по индексу,
* метод, выполняющий необходимые действия в соответствии с вариантом задания (Таблица 1).

Разработать консольное приложение, в котором создаются три объекта-массива. Для создания первого объекта использовать синтаксис инициализатора, для второго данные вводить с клавиатуры, третий объект создать, объединив первые два. Вывести отсортированные массивы. Вывести результаты вычислений в соответствии с индивидуальным заданием для третьего массива.

|  |  |
| --- | --- |
|  | Вычисление произведения отрицательных элементов после первого положительного, вычисление произведения отрицательных элементов после элемента с заданным номером |

Решение:

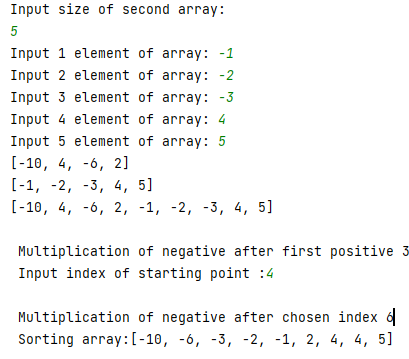
Листинг задания:

package com.company;  
import javax.naming.PartialResultException;  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class LR\_1 {  
  
 private static int[] *array*;  
  
 public LR\_1(int[] arr) {  
 *array* = arr;  
 }  
  
 public String toString() {  
 return Arrays.*toString*(*array*);  
 }  
  
 public static void sorting() {  
 System.*out*.print("\n Sorting array:");  
 Arrays.*sort*(*array*);  
 }  
  
 public void myltiplication(int index)  
 {  
 int multiply = 1;  
 for (int j = index; j < *array*.length; j++)  
 {  
 if (*array*[j] < 0)  
 {  
 multiply\*=*array*[j];  
 }  
  
 }  
 System.*out*.printf("\n Multiplication of negative after first positive %d", multiply);  
 }  
  
 public int indexOfPositive()  
 {  
 int positive = 0;  
 int count = 0;  
  
 for (int i = 0; i < *array*.length;i++)  
 {  
 if (*array*[i] > 0)  
 {  
 count++;  
 if (count == 1)  
 {  
 positive = i;  
 }  
 }  
 }  
 return positive;  
  
 }  
  
 public static void indexMultiplication()  
 {  
 System.*out*.print("\n Input index of starting point :");  
 Scanner in = new Scanner(System.*in*);  
 int number = in.nextInt();  
 int mult = 1;  
 for (int i = number; i < *array*.length; i++)  
 {  
 if (*array*[i] < 0)  
 mult\*= *array*[i];  
 }  
 System.*out*.printf("\n Multiplication of negative after chosen index %d", mult);  
  
 }  
}

Листинг main:

package com.company;  
  
import java.util.Scanner;  
  
public class main {  
  
 public static void main(String[] args) {  
 *test1*();  
 }  
  
 public static void test1() {  
 int[] arrFirst = new int[]{-10, 4, -6, 2};  
 int[] arrSecond = *inputArrSecond*();  
 int[] arrThird = *createThirdArr*(arrFirst, arrSecond);  
  
 LR\_1 arr1 = new LR\_1(arrFirst);  
 System.*out*.println(arr1);  
  
 LR\_1 arr2 = new LR\_1(arrSecond);  
 System.*out*.println(arr2);  
  
 LR\_1 arr3 = new LR\_1(arrThird);  
 System.*out*.println(arr3);  
  
 arr3.myltiplication(arr3.indexOfPositive());  
 arr3.*indexMultiplication*();  
 arr3.*sorting*();  
 System.*out*.println(arr3);  
  
 }  
 public static int[] createThirdArr(int[] arrFirst, int[] arrSecond) {  
 int[] arr = new int[arrFirst.length + arrSecond.length];  
 System.*arraycopy*(arrFirst, 0, arr, 0, arrFirst.length);  
 System.*arraycopy*(arrSecond, 0, arr, arrFirst.length, arrSecond.length);  
  
 return arr;  
 }  
  
 public static int[] inputArrSecond() {  
 System.*out*.println("Input size of second array: ");  
 Scanner in = new Scanner(System.*in*);  
 int sizeOfArr = in.nextInt();  
  
 int[] arr = new int[sizeOfArr];  
  
 for (int index = 0; index < sizeOfArr; index++) {  
 System.*out*.printf("Input %d element of array: ", index + 1);  
 arr[index] = in.nextInt();  
 }  
 return arr;  
 }  
}

Результат:



***Вывод***: Во время выполнения лабораторной работы были получены базовые навыки объектно-ориентированного программирования в среде разработки Java.

# **Приложение A**

**Защита**

***Задание*:** Сравнить быстродействие метода Array.Sort и сортировки пузырьком для массивов размерностью 10, 100, 1000.

**Листинг программы:**

package com.company;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Arrays;

import java.time.Clock;

public class main {

public static void main(String[] args) {

BufferedReader input = new BufferedReader(new InputStreamReader(

System.in));

int i, j, rows, columns, N;

int[][] array = null;

int[][] array2 = null;

try

{

System.out.print("Input row count:\n> ");

rows = Integer.parseInt(input.readLine());

System.out.print("Input column count:\n> ");

columns = Integer.parseInt(input.readLine());

if (rows < 1 || columns < 1)

throw new Exception();

System.out.print("Input range N:\n> ");

N = Integer.parseInt(input.readLine());

if (N < 2)

throw new Exception();

}

catch (IOException e)

{

System.out.println("Program exit with exception.");

return;

}

catch (NumberFormatException e)

{

System.out.println("Program exit with exception.");

return;

}

catch (Exception e)

{

System.out.println("Program exit with error.");

return;

}

array = new int[rows][columns];

array2 = new int[rows][columns];

java.util.Random random = new java.util.Random();

for (i = 0; i < rows; ++i)

for (j = 0; j < columns; ++j)

array[i][j] = random.nextInt(N) + 1;

for (i = 0; i < rows; ++i)

for (j = 0; j < columns; ++j)

array2[i][j] = random.nextInt(N) + 1;

System.out.println("Matrix:");

for (i = 0; i < rows; ++i, System.out.println())

for (j = 0; j < columns; ++j)

System.out.print(array[i][j] + "\t");

System.out.println("Matrix:");

for (i = 0; i < rows; ++i, System.out.println())

for (j = 0; j < columns; ++j)

System.out.print(array2[i][j] + "\t");

long start = System.nanoTime();

for (i = 0; i < rows; i++){

Arrays.sort(array[i]);

}

long timeWorkSort = System.nanoTime() - start;

System.out.println("Array.sort took : " + timeWorkSort + " nanosec");

long start1 = System.nanoTime();

boolean sorted = false;

while (!sorted){

sorted = true;

for (i = 0; i < rows; i++){

for(j = 0; j < columns-1; j++){

if(array2[i][j] > array2[i][j+1])

{

int temp = array2[i][j];

array2[i][j] = array2[i][j+1];

array2[i][j+1] = temp;

sorted = false;

}

}

}

}

long timeWorkSort1 = System.nanoTime() - start1;

System.out.println("Buble sort took : " + timeWorkSort1 + " nanosec");

/\* System.out.print("\n Sorted array: \n");

for (i = 0; i < rows; ++i, System.out.println())

for (j = 0; j < columns; ++j)

System.out.print(array[i][j] + "\t");\*/

}

}