**Министерство науки и высшего образования Российской Федерации**

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н.Ельцина»

ИРИТ-РТФ

Центр ускоренного обучения

**Отчет**

**о лабораторной работе №6**

**«Введение в классы, часть2»**

Руководитель преподаватель Н.А. Архипов

Студент гр. РИЗ-100028у В.А. Кудрявцев

**Екатеринбург 2021**

ЦЕЛЬ РАБОТЫ

Цель выполнения работы введение в работу с классами Java.

ОПИСАНИЕ ЗАДАЧИ

Составить 10 программ по представленным задачам, представить листинги программ, во вложении к отчёту приложить файлы готовых классов.

ХОД ВЫПОЛНЕНИЯ

1. Напишите программу с классом, в котором есть два ноля: символьное и текстовое. В классе должен быть перегруженный метод для присваивания значений нолям. Если метод вызывается с символьным аргументом, то соответствующее значение присваивается символьному полю. Если метод вызывается с текстовым аргументом, то он определяет значение текстового ноля. Методу аргументом также может передаваться символьный массив. Если массив состоит из одного элемента, то он определяет значение символьного поля. В противном случае (если в массиве больше одного элемента) из символов массива формируется текстовая строка и присваивается значением текстовому полю.

import java.util.Scanner;

public class example\_18\_01 {

public static void main(String[] args)

{

Test1 T1 = new Test1();

T1.SET(inputChar());

T1.SET(inputString());

T1.SET(arrayChar());

}

public static String inputString()

{

Scanner input\_value = new Scanner(System.in);

System.out.println("Введите строку: ");

var value = input\_value.next();

return value;

}

public static char inputChar()

{

Scanner input\_value = new Scanner(System.in);

System.out.print("Введите символ: ");

var value = input\_value.next();

return value.charAt(0);

}

public static char[] arrayChar()

{

var value = inputString();

return value.toCharArray();

}

}

class Test1 {

private char c = 0;

private String d = "0";

void SET(char i) {

c = i;

System.out.println("Символьное поле: " + c);

}

void SET(String i) {

d = i;

System.out.println("Строковое поле: " + d);

}

void SET(char[] maschar){

if(maschar.length != 0) c = maschar[0];

else d = String.valueOf(maschar);

System.out.println("Символьное поле: " + c + " Строковое поле: " + d);

}

}

2. Напишите программу с классом, в котором есть закрытое статическое целочисленное ноле с начальным нулевым значением. В классе должен быть описан статический метод, при вызове которого отображается текущее значение статического поля, после чего значение поля увеличивается на единицу.

import java.util.Scanner;

public class example\_18\_02

{

public static void main(String[] args)

{

var i = 0;

var res = valueWhile();

while (i <= res)

{

Test2.addValue();

i++;

}

}

public static int valueWhile ()

{

Scanner input\_value = new Scanner(System.in);

System.out.print("Введите значение с помощью которого вы увидете результат работы класс: ");

var num = input\_value.nextInt();

return num;

}

}

class Test2

{

private static int num = 0;

static void addValue ()

{

System.out.println("Значение статического поля: " + num);

num++;

}

}

3. Напишите программу с классом, в котором есть статические методы, которым можно передавать произвольное количество целочисленных аргументов (или целочисленный массив). Методы, на основании переданных аргументов или массива, позволяют вычислить: наибольшее значение, наименьшее значение, а также среднее значение из набора чисел.

import java.util.Scanner;

public class example\_18\_03

{

public static void main(String[] args)

{

var array= createArray();

Test3.getArray(array);

}

static Integer[] createArray() {

Scanner scanner = new Scanner(System.in);

System.out.print("Введите размер массива: ");

var sizeNums = scanner.nextInt();

Integer[] Nums = new Integer[sizeNums];

System.out.print("Заполните массив собственными интересующими Вас значениями: ");

for (int i = 0; i < Nums.length; i++)

{

Nums[i] = scanner.nextInt();

}

return Nums;

}

}

class Test3 {

public static void getArray(Integer[] array) {

var maxValue = getMaxValue(array);

var minValue = getMinValue(array);

var averageValue = getAverageValue(array);

System.out.println("Максимальное значение набора чисел (массива): " + maxValue + "\n" +

"Минимальное значение набора чисел (массива): " + minValue + "\n" +

"Среднее значение набора чисел (массива): " + averageValue);

}

private static int getAverageValue(Integer[] array) {

int averageValue = 0;

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

averageValue += array[i];

}

return averageValue / array.length;

}

private static int getMinValue(Integer[] array) {

var minValue = array[0];

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

if (minValue > array[i]) minValue = array[i];

}

return minValue;

}

private static int getMaxValue(Integer[] array) {

var maxValue = array[0];

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

if (maxValue < array[i]) maxValue = array[i];

}

return maxValue;

}

}

4. Напишите программу, в которой описан статический метод для вычисления двойного факториала числа, переданного аргументом методу. По определению, двойной факториал числа п (обозначается как n!!) — это произведение через одно всех чисел, не больших числа п. То есть n!! = п \* (n - 2) \* (п - 4)\* ... (последний множитель равен 1 для нечетного п и равен 2 для четного n). Например, 6!! = 6 х 4 х 2 = 48 и 5!! = 5 х 3 х 1 = 15.

import java.util.Scanner;

public class example\_18\_04 {

public static void main(String[] args)

{ Scanner scanner = new Scanner(System.in);

System.out.print("Введите значение вычисляемого двойного факториала: ");

var getValue = Test4.doubleFactorial(scanner.nextInt());

System.out.println("Размер вычисляемого двойного факториала: " + getValue);

}

}

class Test4

{

public static int doubleFactorial(int valueFactorial) {

int fact = 1;

while (valueFactorial >= 1) {

fact \*= valueFactorial;

valueFactorial -= 2;

}

return fact;

}

}

5. Напишите программу со статическим методом, которым вычисляется сумма квадратов натуральных чисел 12 + 22 + 32 + ... + п2. Число п передается аргументом методу. Предложите версию метода с рекурсией и без рекурсии. Для проверки результата можно использовать формулу 12 + 22 +32+…+n2 = (n+l) (2n + 1)/6

import java.util.Scanner;

public class example\_18\_05 {

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

System.out.print("Введите число для вычисления суммы квадратов натуральных чисел: ");

var result = Test5.powValue(scanner.nextInt());

System.out.println("Результат вычисления суммы квадратов натуральных чисел: " + result);

}

}

class Test5

{ static int powValue(int n)

{ int result = 0;

for( int i = 0; i <= n; i++)

{

result += Math.pow(i,2);

}

return result;

}}

6.Напишите программу со статическим методом, которому аргументом передается целочисленный массив и целое число. Результатом метод возвращает ссылку на новый массив, который получается из исходного массива (переданного первым аргументом методу), если в нем взять несколько начальных элементов. Количество элементов, которые нужно

взять из исходного массива, определяются вторым аргументом метода. Если второй аргумент метода больше длины массива, переданного первым аргументом, то методом создается копия исходного массива и возвращается ссылка на эту копию.

import java.util.Scanner;

public class example\_18\_06 {

public static void main(String[] args) {

Scanner InCMD = new Scanner(System.in);

System.out.print("Введите количество элементов для массива = ");

int num = InCMD.nextInt();

int [] x = new int [num];

System.out.print("Созданный массив = ");

for (int i=0;i<num ;i++ ) {

int a = (int) (Math.random()\*(50\_999+1)) - 5\_000;

x[i] = a;

System.out.print(a + " ");

}

Scanner InCMD\_ = new Scanner(System.in);

System.out.print("\nВыберите количество элементов = ");

int num\_ = InCMD\_.nextInt();

System.out.print("Оставшийся массив = ");

example\_18\_06v2 arg\_ = new example\_18\_06v2();

int [] print\_arr = arg\_.Limit(x, num\_);

System.out.print("\nВернувшийся массив = ");

for (int i = 0; i < print\_arr.length; i++){

System.out.print(print\_arr[i] + " ");

}

}

}

public class example\_18\_06v2 {

public static int[] Limit (int[] arr, int lim){

if (lim > arr.length){

lim = arr.length;

}

int[] Num = new int [lim];

for (int i = 0; i < lim; i++){

System.out.print(arr[i] + " ");

Num[i] = arr[i];

}

return Num;

}

}

7. Напишите программу со статическим методом, аргументом которому передастся символьный массив, а результатом возвращается ссылка на целочисленным массив, состоящий из кодов символов из массива- аргумента.

import java.util.Scanner;

public class example\_18\_07 {

public static void main(String[] args) {

System.out.print("Введите строку = ");

Scanner InCMD = new Scanner(System.in);

String str = InCMD.nextLine();

char [] str\_char = str.toCharArray();

System.out.print("Выходной массив = ");

example\_18\_07v2 arg\_ = new example\_18\_07v2();

int [] print\_arr = arg\_.CHAR(str\_char);

for (int i = 0; i < print\_arr.length; i++) {

System.out.print(print\_arr[i] + " ");

}

}

}

public class example\_18\_07v2 {

public static int[] CHAR (char [] ar) {

int [] int\_ar = new int [ar.length];

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

int\_ar [i] = (int) ar [i];

}

return int\_ar;

}

}

8. Напишите программу со статическим методом, аргументом которому передается целочисленный массив, а результатом возвращается среднее значение для элементов массива (сумма значений элементов, деленная на количество элементов в массиве).

import java.util.Scanner;

public class example\_18\_08 {

public static void main(String[] args) {

Scanner InCMD = new Scanner(System.in);

System.out.print("Введите количество элементов для массива = ");

int num = InCMD.nextInt();

int [] x = new int [num];

System.out.print("Созданный массив = ");

for (int i=0;i<num ;i++ ) {

int a = (int) (Math.random()\*(150\_000+1)) - 15\_000;

x[i] = a;

System.out.print(a + " ");

}

example\_18\_08v2 arg\_ = new example\_18\_08v2();

double print\_ = arg\_.SR\_ZNC(x);

System.out.print("\n Среднее значение элемента для массива = " + print\_);

}

}

public class example\_18\_08v2 {

public double SR\_ZNC (int [] ar){

double sr\_znc = 0;

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

sr\_znc += ar[i];

}

sr\_znc /= ar.length;

return sr\_znc;

}

}

9. Напишите программу со статическим методом, аргументом которому передается одномерный символьный массив. В результате вызова метода элементы массива попарно меняются местами: первый — с последним , второй — с предпоследним и так далее.

import java.util.Scanner;

public class example\_18\_09 {

public static void main(String[] args) {

System.out.print("Введите строку = ");

Scanner InCMD = new Scanner(System.in);

String str = InCMD.nextLine();

char [] str\_char = str.toCharArray();

System.out.print("Массив на выходе = ");

example\_18\_09v2 arg\_ = new example\_18\_09v2();

char [] print\_arr = arg\_.MST(str\_char);

for (int i = 0; i < print\_arr.length; i++) {

System.out.print(print\_arr[i] + " ");

}

}}

public class example\_18\_09v2 {

public char[] MST ( char[] a ){

char [] ar = new char [a.length];

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

int buf = a.length - i - 1;

ar[buf] = a[i];

} return ar; }}

10. Напишите программу со статическим методом, аргументом которому передается произвольное количество целочисленных аргументов. Результатом метод возвращает массив из двух элементов: это значения наибольшего и наименьшего значений среди аргументов, переданных методу.

import java.util.Arrays;

import java.util.Scanner;

public class example\_18\_10 {

public static void main(String[] args) {

Scanner InCMD = new Scanner (System.in);

System.out.print("Введите размер массива: ");

int num = InCMD.nextInt();

int [] x = new int [num];

System.out.println("Исходный массив");

for (int i=0;i<num ;i++ ) {

int a = (int) (Math.random()\*(150\_000+1)) - 5\_000;

x[i] = a;

System.out.print(x[i] + " "); }

example\_18\_10v2 AAA = new example\_18\_10v2();

int [] print\_ar = AAA.TF(x);

System.out.print("\nМаксимальный элемент массива = " + print\_ar[1] + "\nМинимальный элемент массива = " + print\_ar[0] );

}

}

public class example\_18\_10v2 {

public int [] TF (int [] a){

int size = a.length - 1;

int [] ar = new int [2];

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

if (a[i] >= 0){

ar[1] = a[i];

}if (a[i] <= 0) {

ar[0] = a[i];

}}

return ar;

}}

ВЫВОДЫ

Написание отдельных методов позволяет объединять классы для слаженной работы при выполнении задач, поставленных пользователем для вычисления.