

**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Уральский государственный университет имени первого президента России  
Б. Н. Ельцина»

Институт радиоэлектроники и информационных технологий  
Центр ускоренного обучения

## **Отчёт по лабораторной работе №6**

**«Введение в классы, часть 2»**

Руководитель ст. преподаватель

Н. А. Архипов

Студент гр. РИЗ-100028у

И. С. Арсентьев

**Екатеринбург 2021**

## Лабораторная работа №6.

### «Введение в классы, часть 2»

*Цель:* 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 задач:

***1. Напишите программу с классом, в котором есть два поля: символьное и текстовое. В классе должен быть перегруженный метод для присваивания значений полям. Если метод вызывается с символьным аргументом, то соответствующее значение присваивается символьному полю. Если метод вызывается с текстовым аргументом, то он определяет значение текстового поля. 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_02_61 {
    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 поле 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.**

```

public class example_02_62
{
    public static void main(String[] args)
    {
        var i = 0;
        var res = valueWhile();
        while (i <= res)
        {
            Test2.addValue();
            i++;
        }
    }
    public static int valueWhile ()//класс для перебора чисел
    {
        Scanner input_value = new Scanner(System.in);

```

```

        System.out.print("Введите значение с помощью которого вы увидите результат работы
класс: ");
        var num = input_value.nextInt();
        return num;
    }
}
class Test2
{
    private static int num = 0;
    //вывод на экран массива чисел от 0 до введенного значения
    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_02_63
{
    public static void main(String[] args)
    {
        var array= createArray();
        Test3.getArray(array);
    }
    static Integer[] createArray() {
        Scanner scanner = new Scanner(System.in);
        // ввод пользователем размера массива.
        System.out.print("Введите размер массива: ");
        var sizeNums = scanner.nextInt();
        Integer[] Nums = new Integer[sizeNums];
        System.out.print("Заполните массив собственными интересующими Вас значениями: ");
        // заполните массив собственными интересующими значениями.
        for (int i = 0; i < Nums.length; i++)
        {
            Nums[i] = scanner.nextInt();
        }
        return Nums;
    }
}
class Test3
{
    //основной метод
    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)//метод поиска минимального значения в массиве
{
    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)//метод поиска максимального значения в массиве
{
    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  $n$  (обозначается как  $n!!$ ) — это произведение через одно всех чисел, не больших числа  $n$ . То есть  $n!! = n * (n - 2) * (n - 4) * \dots$  (последний множитель равен 1 для нечетного  $n$  и равен 2 для четного  $n$ ). Например,  $6!! = 6 \times 4 \times 2 = 48$  и  $5!! = 5 \times 3 \times 1 = 15$ . Предложите версию метода без рекурсии и с рекурсией.**

```

import java.util.Scanner;
public class example_02_64 {
    public static void main(String[] args)//основной метод класса
    {
        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)//метод поиска двойного факториала
    {
        int fact = 1;
        while (valueFactorial >= 1) {
            fact *= valueFactorial;
            valueFactorial -= 2;
        }
        return fact;//возврат значения fact в основной метод
    }
}

```

**5. Напишите программу со статическим методом, которым вычисляется сумма квадратов натуральных чисел  $1^2 + 2^2 + 3^2 + \dots + n^2$ . Число  $n$  передается аргументом методу. Предложите версию метода с рекурсией и без рекурсии. Для проверки результата можно использовать формулу  $1^2 + 2^2 + 3^2 + \dots + n^2 = (n+1)(2n+1)/6$**

```

import java.util.Scanner;
public class example_02_65 {
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);//создание объекта сканера введенного
        значения с клавиатуры пользователем
        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++)//перебор натуральных чисел
        {
            result += Math.pow(i,2);//возведение числа в квадрат и суммирование с предыдущей
            суммой
        }
        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_02_66 {
    public static void main(String[] args) {
        Scanner InCMD = new Scanner(System.in); //создание объекта Scanner для ввода значения с
        клавиатуры
        System.out.print("Введите количество элементов для массива = ");
        int num = InCMD.nextInt();
        //Создание массива случайных чисел для выборки
        int [] x = new int [num];
        System.out.print("Созданный массив = ");
        //генерация массива,заполнение случайными числами
        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 = "); //выбор количества элементов
        для вывода
        int num_ = InCMD_.nextInt();

        System.out.print("Оставшийся массив = ");
        example_02_66v2 arg_ = new example_02_66v2();
        int [] print_arr = arg_.Limit(x, num_);
        System.out.print("\nВернувшийся массив = "); //вывод вернувшегося массива из
        отдельного класса
        for (int i = 0; i < print_arr.length; i++){
            System.out.print(print_arr[i] + " ");
        }
    }
}

public class example_02_66v2 {
    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 передается символьный массив, а результатом возвращается ссылка на целочисленный массив, состоящий из кодов символов из массива- аргумента.**

```

import java.util.Scanner;
public class example_02_67 {
    public static void main(String[] args) {
        System.out.print("Введите строку = "); //строка диалог с пользователем
        Scanner InCMD = new Scanner(System.in);
        String str = InCMD.nextLine();
        char [] str_char = str.toCharArray(); //инициализация элементов массива, введенного
        пользователем
        System.out.print("Выходной массив = "); //на экран выводится массив, состоящий из кодов
        введенных символов
        example_02_67v2 arg_ = new example_02_67v2(); //массив формируется в отдельном
        классе
        int [] print_arr = arg_.CHAR(str_char);
        for (int i = 0; i < print_arr.length; i++) {
            System.out.print(print_arr[i] + " ");
        }
    }
}

```

//массив, формирующийся в отдельном классе

```

public class example_02_67v2 {
    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_02_68 {
    public static void main(String[] args) {

```



```

Scanner InCMD = new Scanner(System.in);
System.out.print("Введите количество элементов для массива = "); //сообщение
пользователю на экране для диалога
int num = InCMD.nextInt();
//Создание массива случайных чисел из диапазона для выборки
int [] x = new int [num];
System.out.print("Созданный массив = ");
//случайное наполнение массива
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_02_68v2 arg_ = new example_02_68v2();//ссылка на массив,создаваемый в другом
классе
double print_ = arg_.SR_ZNC(x);
System.out.print("\n Среднее значение элемента для массива = " + print_);
}
}

```

```

//создание массива в отдельном классе
public class example_02_68v2 {
    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;//возврат среднего значения элемента массива
    }
}

```

**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_02_69 {
    public static void main(String[] args) {
        System.out.print("Введите строку = "); //строка для пользователя на экране
        Scanner InCMD = new Scanner(System.in);
        String str = InCMD.nextLine();
        char [] str_char = str.toCharArray();//наполнение массива символами со введенной строки
        System.out.print("Массив на выходе = ");
        example_02_69v2 arg_ = new example_02_69v2();//ссылка на массив в сопряженном классе
        char [] print_arr = arg_.MST(str_char);
        for (int i = 0; i < print_arr.length; i++) {
            System.out.print(print_arr[i] + " ");
        }
    }
}

```

```
//массив, в котором собираются элементы перебираемого массива
public class example_02_69v2 { //массив в отдельном классе для перебора элементов
введённой строки
    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_02_70 {
    public static void main(String[] args) {
        Scanner InCMD = new Scanner (System.in);
        System.out.print("Введите размер массива: "); //диалог с пользователем
        int num = InCMD.nextInt();
        int [] x = new int [num];
        System.out.println("Исходный массив"); //вывод массива на экран
        for (int i=0;i<num ;i++ ) { //наполнение массива случайными числами из промежутка
            int a = (int) (Math.random()*(150_000+1)) - 5_000;
            x[i] = a;
            System.out.print(x[i] + " ");
        }
        example_02_70v2 AAA = new example_02_70v2();
        int [] print_ar = AAA.TF(x);
        System.out.print("\nМаксимальный элемент массива = " + print_ar[1] + "\nМинимальный
элемент массива = " + print_ar[0] ); //вывод результата работы на экран
    }
}
public class example_02_70v2 {
    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; //возврат значения переменной в метод
    }
}
```

***Вывод:***

Написание отдельных методов позволяет объединять классы для сложной работы при выполнении задач, поставленных пользователем для вычисления. Работа всех программ проверена, в архиве соответствующие классы будут прикреплены.