**ПРИЛОЖЕНИЕ Е**

**Выполнение работ по разработке ядра программы согласно проектной документации**

**Цель работы:** приобрести умения разработки программы согласно требованиям проектной документации

**Оборудование:** ПК, IntelliJ IDEA

Источники:

1. Руководство Java. <https://metanit.com/java/tutorial/>
2. [Массивы Java: объявление, создание, заполнение и инициализация, определение размера, вывод на экран, методы и примеры (sky.pro)](https://sky.pro/media/massivy-java/?ysclid=lpplhkykdq723199542)
3. [Методы Arrays fill, copyOf, copyOfRange, sort, binarySearch в Java (javarush.com)](https://javarush.com/quests/lectures/questsyntaxpro.level05.lecture07)
4. [Класс Arrays в Java (javarush.com)](https://javarush.com/groups/posts/1933-klass-arrays-i-ego-ispoljhzovanie)
5. Введение в массивы. <https://youtu.be/_mGgR5OhI6I>
6. Экземплярные и статические методы. <https://youtu.be/RVckAdjY1Yk>

**ЗАДАНИЯ К ПРАКТИЧЕСКОМУ ЗАНЯТИЮ**

В ИП «Программист» поступил заказ на разработку приложения для сотрудников метеостанции, которая ежедневно С 8 до 20 часов получает температуру воздуха ежечасно с регистратора температуры (генератора случайных чисел). Регистратор в течение суток данные записывает в одномерный массив. Ваша задача разработать класс Library, класс должен содержать пять методов:

1. Вывода на экран максимальной и минимальной температуру.
2. Определял сколько раз температура в течение суток достигала минимума и максимума
3. Определял среднюю температуру воздуха в течение суток
4. Определял в котором часу была впервые отмечена минимальная/максимальная температура.
5. Определял интервал времени между временем регистрации минимальной и максимальной температуры
6. Определял, встречалась ли в течение дня нулевая температура

Класс Library, класс должен быть построен с соблюдением следующих требований к методам:

1. Метод первый – статический, проверяющий правильность расчета размерности массива;
2. Метод второй – статический, инициализирующий массив введенной с клавиатуры размерности случайными целыми числами и возвращающий в качестве значения результат его заполнения;
3. Метод четвертый – экземплярный, возвращающий значение через свойство класса-библиотеки (для метода тип возвращающего значения – void);
4. Метод пятый – экземплярный, геттер для возвращения результата работы четвертого метода.
5. Остальные методы обработки – экземплярные, реализующий основное задание списка индивидуальных заданий и возвращающий значение с помощью ключевого слова return.

ОТЧЕТ

**Класс Main**

import java.util.Arrays;  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String args[]) {  
 Scanner in = new Scanner(System.*in*);  
 int n;  
 do {  
 System.*out*.println("введите натуральное число:");  
 n = in.nextInt();  
 }  
 while (lib.*conditionVerify*(n) == false);  
 in.close();  
 final int MIN = 0;  
 final int MAX = 10;  
 int[] a = lib.*generationArray*(MIN, MAX, n);  
 System.*out*.println(Arrays.*toString*(a));  
 lib obj = new lib();  
 obj.maxIndex(a);  
 int[] minmax = obj.method1(a);  
  
 System.*out*.println("Минимумымым " + minmax[0] + "\n Индекс последнего максиманального " + obj.getIndMax() + "\n максиманальное значение " + minmax[1]);  
  
 int[] minbmax = obj.minmax(a);  
 System.*out*.println("миниманальное было столько " + minbmax[0] + " \n но максиманальное было столько " + minbmax[1]);  
  
 double sredniy = obj.sredniy(a);  
 System.*out*.println("среднееЕЕЕЕееЕЕ число " + sredniy);  
  
 int[] timeminmax = obj.timeminmax(a);  
 System.*out*.println("миниманальная " + timeminmax[0] + " максиманальная " + timeminmax[1]);  
  
 int prom = obj.prom(a);  
 System.*out*.println("промежуток " + prom);  
  
 int nol = obj.nol(a);  
 System.*out*.println("Ноликов-еболиков столько " + nol);

**Класс Library**

public class lib {  
 private int indMax=0;  
 public static boolean conditionVerify(int n){  
 if (n<=0){  
 System.*out*.println("Овощной деградант, число больше 0 .\n"+" попробуй нормально ввести :( \n");  
 return false;}  
 return true;}  
 public static int [] generationArray(final int MIN, final int MAX, int n){  
 int [] array= new int[n];  
 for (int i=0; i< array.length; i++){  
 array[i]=(int)(Math.*random*()\*(MAX-MIN)+MIN);  
 }  
 return array;  
  
 }  
 public int min( int [] array ){  
 int min=array [0];  
 for (int i=1; i< array.length; i++){  
 if (min > array[i]) min =array[i];  
 }  
 return min;  
 }  
  
 public int max( int [] array ){  
 int max=array [0];  
 for (int i=1; i< array.length; i++ ){  
 if (max < array[i]) max =array[i];  
 }  
 return max;  
 }  
 public void maxIndex(int [] array ){  
 for (int i=1; i< array.length; i++){  
 if(array[indMax]<=array[i]) indMax=i;  
 }  
 }  
 public int getIndMax(){  
 return indMax;  
 }  
 public int[] method1(int [] arr){  
 return new int[]{this.min(arr),this.max(arr)};  
 }  
  
 public int[] minmax (int [] array) {  
 int minb = min(array);  
  
 int max = max(array);  
  
 int c\_min = 0;  
 int c\_max = 0;  
  
 for (int i = 0; i < array.length; i++) {  
 if (minb == array[i]) c\_min = c\_min + 1;  
 if (max==array[i]) c\_max=c\_max+1;  
  
 }  
 return new int[]{c\_min,c\_max};  
 }  
  
 public double sredniy (int []array ){  
 int s=0;  
 for (int i = 0; i < array.length; i++) {  
 s=s+array[i];  
 }  
 return (double) s /array.length;  
 }  
 public int[] timeminmax (int[] array ) {  
 int timemin = min(array);  
 int timemax = max(array);  
  
 for (int i = 0; i < array.length; i++) {  
 if (timemin == array[i]) {  
 timemin = i+1;  
 break;  
 }  
 }  
 for (int i = 0; i < array.length; i++) {  
 if (timemax == array[i]) {  
 timemax = i+1;  
 break;  
 }  
  
 }  
 return new int[]{timemin, timemax};  
 }  
 public int prom (int[] array ) {  
 int a= timeminmax(array)[0];  
 int b= timeminmax(array)[1];  
 return Math.*abs*(a-b);  
 }  
  
  
 public int nol (int []array){  
 int n=0;  
 for (int i = 0; i < array.length; i++) {  
 if (0 == array[i]) {  
 n=n+1;  
  
 }  
  
 }  
 return n;  
  
 }  
}

**Вывод: Я** приобрел умения разработки программы согласно требованиям проектной документации.