**ПРИЛОЖЕНИЕ**

**Выполнение работ по разработке функциональной части программы**

**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 Температура **(**его возможные свойства: дата и время измерения, температура в цельсиях, в кельвинах, фаренгейтах, влажность воздуха, наличие осадков**)**; в классе Температура в соответствии со структурой его свойств необходимо переопределить метод toString() суперкласса Object; переопределить разработанные методы в прошлом занятии с новым массивом.

Разработать интерфейс Applicable с тремя абстрактными публичными экземплярными методами:

* для проверки корректности ввода размерности;
* для создания массива объектов заданной с клавиатуры размерности;
* для вывода массива объектов на экран.

Разработанный ранее класс Library должен реальзовать интерфейс Applicable, а кроме того, имеющий статический метод выбора элемента (элементов) из массива по определенному критерию.

Класс Мэйн

import java.util.Scanner;  
  
public class Main  
{  
 public static void main (String args []){  
 Library obj= new Library();  
 Scanner in=new Scanner(System.*in*);  
 int n;  
 do {  
 System.*out*.println("введите натуральное n:");  
 n=Integer.*parseInt*(in.nextLine());  
 }  
 while (obj.conditionVerify(n)==false);  
 temp[] i= obj.generationArray(n);  
 System.*out*.println("исходная база данных :");  
 obj.display(i);  
 System.*out*.println("введите нижнию границу температуры:");  
 double minTemp =Integer.*parseInt*(in.nextLine());  
 System.*out*.println("введите нижнию границу температуры");  
 double maxTemp =Integer.*parseInt*(in.nextLine());  
 in.close();  
 System.*out*.println("результат выбора");  
 Library.*resultChoise*(i,minTemp,maxTemp);  
  
 }  
}

Интерфейс Апликабле

package library;  
  
public interface Applicable {  
 abstract public boolean conditionVeryfy(int n);  
 abstract public temperatyra[] generationArray (int n);  
 public void display (temperatyra[] array);  
}

Класс Либ

import java.util.Date;  
import java.util.Scanner;  
  
public class Library implements Applicable{  
  
 @Override  
 public boolean conditionVerify(int n) {  
 if (n <= 0) {  
 System.*out*.println("\n n должно быть больше 0."+ "\n " +  
 "попробуйте еще раз");  
 return false;  
  
 }  
 return true;  
 }  
  
 @Override  
 public temp[] generationArray(int n) {  
 Scanner in=new Scanner(System.*in*);  
 temp[] array=new temp[n];  
 for (int i = 0; i < n; i++) {  
 System.*out*.println("Ввод "+ (i+1) +" записи");  
 System.*out*.println("\t Цельсий");  
 double Tc=in.nextDouble();  
 System.*out*.println("\t Фаренгейт");  
 double Tf=in.nextDouble();  
 System.*out*.println("\t Кельвин");  
 double Tk= in.nextDouble();  
 System.*out*.println("\t Осадки");  
 double Rain= in.nextDouble();  
  
 System.*out*.println("\t Дата(год,месяц,день )");  
 int год=in.nextInt();  
 int месяц=in.nextInt();  
 int день=in.nextInt();  
  
 Date Dota=new Date(год,месяц,день);  
  
 System.*out*.println("\tВлажность");  
 double humidity = in.nextDouble();  
 array[i]= new temp(Tc,Tf,Rain,Tk,Dota,humidity);  
 }  
 return array ;  
 }  
  
 @Override  
 public void display(temp[] array) {  
 for (int i = 0; i < array.length; i++) {  
 System.*out*.println(array[i].toString());  
  
 }  
  
 }  
 public static void resultChoise(temp[] array, double minTemp, double maxTemp){  
 double temp;  
 for (int i = 0; i < array.length; i++) {  
 temp=array[i].getTc();  
 if (minTemp<=temp && temp<=maxTemp)  
 System.*out*.println(array[i].toString());  
  
 }  
 }  
}

Класс температура

import java.util.Date;  
  
*/\*\**  
 *\* Class temperatyra*  
 *\*/*  
public class temperatyra {  
  
 //  
 // Fields  
 //  
  
 private double celsi;  
 private double kelvin;  
 private double falengeit;  
 private double vlazhnost;  
 private double osadki;  
 private Date data;  
   
 //  
 // Constructors  
 //  
 public temperatyra (double celsi, double falengeit, double osadki, double  
 kelvin, Date data, double vlazhnost) {  
 this.celsi= celsi;  
 this.falengeit=falengeit;  
 this.kelvin=kelvin;  
 this.data=data;  
 this.vlazhnost=vlazhnost;  
  
 };  
   
 //  
 // Methods  
 //  
  
  
  
  
 //  
 // Accessor methods  
 //  
  
 */\*\**  
 *\* Set the value of celsi*  
 *\* @param newVar the new value of celsi*  
 *\*/*  
public void setCelsi (double newVar) {  
 celsi = newVar;  
 }  
  
 */\*\**  
 *\* Get the value of celsi*  
 *\* @return the value of celsi*  
 *\*/*  
public double getCelsi () {  
 return celsi;  
 }  
  
 */\*\**  
 *\* Set the value of kelvin*  
 *\* @param newVar the new value of kelvin*  
 *\*/*  
public void setKelvin (double newVar) {  
 kelvin = newVar;  
 }  
  
 */\*\**  
 *\* Get the value of kelvin*  
 *\* @return the value of kelvin*  
 *\*/*  
public double getKelvin () {  
 return kelvin;  
 }  
  
 */\*\**  
 *\* Set the value of falengeit*  
 *\* @param newVar the new value of falengeit*  
 *\*/*  
public void setFalengeit (double newVar) {  
 falengeit = newVar;  
 }  
  
 */\*\**  
 *\* Get the value of falengeit*  
 *\* @return the value of falengeit*  
 *\*/*  
public double getFalengeit () {  
 return falengeit;  
 }  
  
 */\*\**  
 *\* Set the value of vlazhnost*  
 *\* @param newVar the new value of vlazhnost*  
 *\*/*  
public void setVlazhnost (double newVar) {  
 vlazhnost = newVar;  
 }  
  
 */\*\**  
 *\* Get the value of vlazhnost*  
 *\* @return the value of vlazhnost*  
 *\*/*  
public double getVlazhnost () {  
 return vlazhnost;  
 }  
  
 */\*\**  
 *\* Set the value of osadki*  
 *\* @param newVar the new value of osadki*  
 *\*/*  
public void setOsadki (double newVar) {  
 osadki = newVar;  
 }  
  
 */\*\**  
 *\* Get the value of osadki*  
 *\* @return the value of osadki*  
 *\*/*  
public double getOsadki () {  
 return osadki;  
 }  
  
 */\*\**  
 *\* Set the value of data*  
 *\* @param newVar the new value of data*  
 *\*/*  
public void setData (Date newVar) {  
 data = newVar;  
 }  
  
 */\*\**  
 *\* Get the value of data*  
 *\* @return the value of data*  
 *\*/*  
public Date getData () {  
 return data;  
 }  
  
  
  
 //  
 // Other methods  
 //  
  
 public String toString (){  
 return "Цельсий"+ this.getCelsi()+"фаленгейт "+ this.getFalengeit()+"."+"кельвины"  
 + this.getKelvin()+"дата "+this.getData();  
 }  
}