КЛАСС AUTOBUS

/\*  
Vorobyov Alexander Olegovich,  
1032193998, NFIbd-01-19  
-----------------------------  
11. Autobus содержащий следующие поля: название начального пункта;  
название конечного пункта; номер маршрута; время поездки  
 \*/  
import java.util.Scanner;  
public class Autobus {  
 private String StartPoint, FinalPoint; //переменные для назв. нач. и кон. пунктов  
 private int RouteNum; //переменные с номером маршрута  
 private double TravelTime; //и его длительностью  
  
 public Autobus(String SP, String FP, int RN, double TT) { //конструктор1  
 this.StartPoint = SP;  
 this.FinalPoint = FP;  
 this.RouteNum = RN;  
 this.TravelTime = TT;  
 }  
  
 public Autobus() { //конструктор2  
 this.StartPoint = "Moscow";  
 this.FinalPoint = "Kazan";  
 this.RouteNum = 1015;  
 this.TravelTime = 12.15;  
 }  
  
 static void TimeDifference(double TT1, double TT2) { //разница в продолжит. поездок  
 double TD = Math.*abs*(TT1 - TT2); //вычисляем модуль разницы  
 System.*out*.println("Difference in trip: " + TD + " hour(s)");  
 }  
  
 static void ShortestTrip(double TT1, double TT2, double TT3, double TT4, double TT5, double TT6) {  
 //поиск самой короткой по времени поездки  
 double min = 1000; //переменная для минимума  
 double[] array; //массив с данными о длитльности поездок  
 array = new double[6];  
 array[0] = TT1; //присваиваем данные  
 array[1] = TT2;  
 array[2] = TT3;  
 array[3] = TT4;  
 array[4] = TT5;  
 array[5] = TT6;  
 for(int i = 0; i < 6; i++) { //через цикл находим кратчайшую поездку  
 if (array[i] < min) {  
 min = array[i];  
 }  
 }  
 System.*out*.println("The shortest trip lasts " + min + " hour(s)");  
 }  
  
 public static void main(String[] args) {  
  
 //заполнение данных об автобусах  
 Autobus autobus1 = new Autobus("Moscow", "Kiev", 560, 10.9);  
 Autobus autobus2 = new Autobus("Moscow", "Minsk", 915, 7.9);  
 Autobus autobus3 = new Autobus("Moscow", "Sevastopol", 440, 20.5);  
 Autobus autobus4 = new Autobus("Moscow", "Saint-Petersburg", 685, 7.5);  
 Autobus autobus5 = new Autobus("Moscow", "Arkhangelsk", 590, 7.4);  
 Autobus autobus6 = new Autobus();  
  
 *TimeDifference*(autobus1.TravelTime, autobus2.TravelTime); //разница во времени между поездками  
 *ShortestTrip*(autobus1.TravelTime, autobus2.TravelTime, autobus3.TravelTime, //самая короткая по времени поездка  
 autobus4.TravelTime, autobus5.TravelTime, autobus6.TravelTime);  
 }  
}

РЕЗУЛЬТАТ

Изображение выглядит как текст

Автоматически созданное описание

КЛАСС TABLE

/\*  
Vorobyov Alexander Olegovich,  
1032193998, NFIbd-01-19  
-----------------------------  
36. Table содержащий следующие поля: страна производитель стола;  
высота стола; материал, из которого изготовлен; цена.  
 \*/  
import java.util.Scanner;  
public class Table {  
 private String Country, Material; //переменные  
 private double Height, Price;  
  
 public Table(String C, String M, double H, double P) { //конструктор1  
 this.Country = C;  
 this.Material = M;  
 this.Height = H;  
 this.Price = P;  
 }  
  
 public Table() { //конструктор2  
 this.Country = "Russia";  
 this.Material = "Chipboard";  
 this.Height = 75;  
 this.Price = 16190;  
 }  
  
 static void FindDomestic(String C1, String C2, String C3, //поиск столов отечественного производства  
 String C4, String C5, String C6) {  
 String[] array;  
 array = new String[6]; //массив с данными о стране производителя  
 array[0] = C1; //присваиваем данные  
 array[1] = C2;  
 array[2] = C3;  
 array[3] = C4;  
 array[4] = C5;  
 array[5] = C6;  
 for(int i = 0; i < 6; i++) { // находим совпадения со страной производства Russia  
 if (array[i] == "Russia") {  
 System.out.println("Table " + (i+1) + " is domestic");  
 }  
 }  
 }  
 static void FindTallest(double H1, double H2, double H3, //поиск самого высокого стола  
 double H4, double H5, double H6) {  
 double max = 0; //переменная для записи максимума  
 double[] array2; //массив с данными о высоте столов  
 array2 = new double[6];  
 array2[0] = H1; //присваиваем данные  
 array2[1] = H2;  
 array2[2] = H3;  
 array2[3] = H4;  
 array2[4] = H5;  
 array2[5] = H6;  
 for (int j = 0; j < 6; j++) { // находим максимальное значение  
 if (array2[j] > max) {  
 max = array2[j];  
 }  
 }  
 System.out.println("The tallest table is " + max + " cm");  
 }  
  
 public static void main(String[] args) {  
 //заполнение данных о столах  
 Table table1 = new Table("Russia", "Chipboard", 75.1, 7490);  
 Table table2 = new Table("China", "Aluminum", 47, 6999);  
 Table table3 = new Table("Sweden", "Wood", 70, 14590);  
 Table table4 = new Table("China", "Glass", 65.5, 8999);  
 Table table5 = new Table("Sweden", "Rock", 84, 19990);  
 Table table6 = new Table();  
  
 FindDomestic(table1.Country, table2.Country, table3.Country, //Стол(ы) отечественного  
 table4.Country, table5.Country, table6.Country); //производства  
 FindTallest(table1.Height, table2.Height, table3.Height, //Самый высокий стол  
 table4.Height, table5.Height, table6.Height);  
 }  
}

РЕЗУЛЬТАТ

Изображение выглядит как текст

Автоматически созданное описание