package com.company;  
  
class Trip {  
 private String StartPoint, FinalPoint; //переменные для назв. нач. и кон. пунктов  
 private int RouteNum; //переменные с номером маршрута  
 private double TravelTime, RouteLen, TicketPrice; //его длительностью, длиной, ценой билета.  
 public static int *n* = 6;  
  
 public Trip(String SP, String FP, int RN, double TT, double RL, double TP) { //конструктор1  
 this.StartPoint = SP;  
 this.FinalPoint = FP;  
 this.RouteNum = RN;  
 this.TravelTime = TT;  
 this.RouteLen = RL;  
 this.TicketPrice = TP;  
 }  
  
 public Trip() { //конструктор2  
 this.StartPoint = "Moscow";  
 this.FinalPoint = "Kazan";  
 this.RouteNum = 1015;  
 this.TravelTime = 12.15;  
 this.RouteLen = 820;  
 this.TicketPrice = 1703;  
 }  
  
 static void TimeDifference(Trip a1, Trip a2) { //разница в продолжит. поездок  
 double TD = Math.*abs*(a1.TravelTime - a2.TravelTime); //вычисляем модуль разницы  
 System.*out*.println("Difference in trip to " + a1.FinalPoint + " and to " + a2.FinalPoint + ": " + TD + " hour(s)");  
 }  
  
 static void ShortestTrip(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 //поиск самой короткой по времени поездки  
 double min = 1000; //переменная для минимума  
 double[] array; //массив с данными о длительности поездок  
 array = new double[*n*];  
 array[0] = a1.TravelTime; //присваиваем данные  
 array[1] = a2.TravelTime;  
 array[2] = a3.TravelTime;  
 array[3] = a4.TravelTime;  
 array[4] = a5.TravelTime;  
 array[5] = a6.TravelTime;  
 for(int i = 0; i < *n*; i++) { //через цикл находим кратчайшую поездку  
 if (array[i] < min) {  
 min = array[i];  
 }  
 }  
 System.*out*.println("The shortest trip lasts " + min + " hour(s)");  
 }  
  
 static void LongestTrip(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 //поиск самой длинной по времени поездки  
 double max = 0; //переменная для минимума  
 double[] array; //массив с данными о длительности поездок  
 array = new double[*n*];  
 array[0] = a1.TravelTime; //присваиваем данные  
 array[1] = a2.TravelTime;  
 array[2] = a3.TravelTime;  
 array[3] = a4.TravelTime;  
 array[4] = a5.TravelTime;  
 array[5] = a6.TravelTime;  
 for(int i = 0; i < *n*; i++) { //через цикл находим кратчайшую поездку  
 if (array[i] > max) {  
 max = array[i];  
 }  
 }  
 System.*out*.println("The longest trip lasts " + max + " hour(s)");  
 }  
  
 static void InfoAboutRoutes(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 //печать информации о направлении маршрута и его номера  
 String[] destination; //массив с пунктами назначения  
 destination = new String[*n*];  
 destination[0] = a1.FinalPoint; destination[1] = a2.FinalPoint; destination[2] = a3.FinalPoint;  
 destination[3] = a4.FinalPoint; destination[4] = a5.FinalPoint; destination[5] = a6.FinalPoint;  
  
 int[] numbers; //массив с номерами маршрутов  
 numbers = new int[*n*];  
 numbers[0] = a1.RouteNum; numbers[1] = a2.RouteNum; numbers[2] = a3.RouteNum;  
 numbers[3] = a4.RouteNum; numbers[4] = a5.RouteNum; numbers[5] = a6.RouteNum;  
  
 for (int j = 0; j < *n*; j++) { //печатаем по шаблону  
 System.*out*.println("Bus number " + numbers[j] + " goes to " + destination[j] + ".");  
 }  
 }  
  
 static void AverageSpeed(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 String[] destination; //массив с пунктами назначения  
 destination = new String[*n*]; // заполняем значения  
 destination[0] = a1.FinalPoint; destination[1] = a2.FinalPoint; destination[2] = a3.FinalPoint;  
 destination[3] = a4.FinalPoint; destination[4] = a5.FinalPoint; destination[5] = a6.FinalPoint;  
  
 double[] time; //массив с длительностью поездки  
 time = new double[*n*];  
 time[0] = a1.TravelTime; time[1] = a2.TravelTime; time[2] = a3.TravelTime;  
 time[3] = a4.TravelTime; time[4] = a5.TravelTime; time[5] = a6.TravelTime;  
  
 double[] len; //массив с длиной маршрута  
 len = new double[*n*];  
 len[0] = a1.RouteLen; len[1] = a2.RouteLen; len[2] = a3.RouteLen;  
 len[3] = a4.RouteLen; len[4] = a5.RouteLen; len[5] = a6.RouteLen;  
  
 for (int a = 0; a < *n*; a++) {  
 double tmp = len[a] / time[a]; //вычисление средней скорости  
 System.*out*.println("Average bus speed to " + destination[a] + ": " + Math.*ceil*(tmp) + " km/h");  
 }  
 }  
  
 static void LongestRoute(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 double[] len; //массив с длиной маршрута  
 len = new double[*n*];  
 len[0] = a1.RouteLen; len[1] = a2.RouteLen; len[2] = a3.RouteLen;  
 len[3] = a4.RouteLen; len[4] = a5.RouteLen; len[5] = a6.RouteLen;  
  
 int[] numbers; //массив с номерами маршрутов  
 numbers = new int[*n*];  
 numbers[0] = a1.RouteNum; numbers[1] = a2.RouteNum; numbers[2] = a3.RouteNum;  
 numbers[3] = a4.RouteNum; numbers[4] = a5.RouteNum; numbers[5] = a6.RouteNum;  
  
 double max = 0;  
 int num = 0;  
  
 for (int e = 0; e < *n*; e++) { //находим максимальное значение  
 if (len[e] > max) {  
 max = len[e];  
 num = numbers[e];  
 }  
 }  
  
 System.*out*.println("The longest route is by bus " + num + ": " + max + " km.");  
 }  
  
 static void ShortestRoute(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 double[] len; //массив с длиной маршрута  
 len = new double[*n*];  
 len[0] = a1.RouteLen; len[1] = a2.RouteLen; len[2] = a3.RouteLen;  
 len[3] = a4.RouteLen; len[4] = a5.RouteLen; len[5] = a6.RouteLen;  
  
 int[] numbers; //массив с номерами маршрутов  
 numbers = new int[*n*];  
 numbers[0] = a1.RouteNum; numbers[1] = a2.RouteNum; numbers[2] = a3.RouteNum;  
 numbers[3] = a4.RouteNum; numbers[4] = a5.RouteNum; numbers[5] = a6.RouteNum;  
  
 double min = 1000;  
 int num = 0;  
  
 for (int e = 0; e < *n*; e++) { //находим максимальное значение  
 if (len[e] < min) {  
 min = len[e];  
 num = numbers[e];  
 }  
 }  
  
 System.*out*.println("The shortest route is by bus " + num + ": " + min + " km.");  
 }  
  
 static void GetPrice(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 double[] price; //массив с ценами на поездки  
 price = new double[*n*]; //заполняем значения  
 price[0] = a1.TicketPrice; price[1] = a2.TicketPrice; price[2] = a3.TicketPrice;  
 price[3] = a4.TicketPrice; price[4] = a5.TicketPrice; price[5] = a6.TicketPrice;  
  
 String[] destination; //массив с пунктами назначения  
 destination = new String[*n*]; // заполняем значения  
 destination[0] = a1.FinalPoint; destination[1] = a2.FinalPoint; destination[2] = a3.FinalPoint;  
 destination[3] = a4.FinalPoint; destination[4] = a5.FinalPoint; destination[5] = a6.FinalPoint;  
  
 for (int l = 0; l < *n*; l++) {  
 System.*out*.println("A bus ticket to " + destination[l] + " costs " + price[l] + " Rub.");  
 }  
 }  
  
 static void StudentSale30(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 double[] price; //массив с ценами на поездки  
 price = new double[*n*]; //заполняем значения  
 price[0] = a1.TicketPrice; price[1] = a2.TicketPrice; price[2] = a3.TicketPrice;  
 price[3] = a4.TicketPrice; price[4] = a5.TicketPrice; price[5] = a6.TicketPrice;  
  
 String[] destination; //массив с пунктами назначения  
 destination = new String[*n*]; // заполняем значения  
 destination[0] = a1.FinalPoint; destination[1] = a2.FinalPoint; destination[2] = a3.FinalPoint;  
 destination[3] = a4.FinalPoint; destination[4] = a5.FinalPoint; destination[5] = a6.FinalPoint;  
  
 for (int l = 0; l < *n*; l++) {  
 double saleprice = price[l] \* 0.7;  
 System.*out*.println("Discount ticket price to " + destination[l] + " for students: " + Math.*ceil*(saleprice) + " Rub.");  
 }  
 }  
  
 static void ChildTariff50(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 double[] price; //массив с ценами на поездки  
 price = new double[*n*]; //заполняем значения  
 price[0] = a1.TicketPrice; price[1] = a2.TicketPrice; price[2] = a3.TicketPrice;  
 price[3] = a4.TicketPrice; price[4] = a5.TicketPrice; price[5] = a6.TicketPrice;  
  
 String[] destination; //массив с пунктами назначения  
 destination = new String[*n*]; // заполняем значения  
 destination[0] = a1.FinalPoint; destination[1] = a2.FinalPoint; destination[2] = a3.FinalPoint;  
 destination[3] = a4.FinalPoint; destination[4] = a5.FinalPoint; destination[5] = a6.FinalPoint;  
  
 for (int l = 0; l < *n*; l++) {  
 double saleprice = price[l] \* 0.5;  
 System.*out*.println("Discount ticket price to " + destination[l] + " for children: " + Math.*ceil*(saleprice) + " Rub.");  
 }  
 }  
  
 static void CheapestTrip(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 double[] price; //массив с ценами на поездки  
 price = new double[*n*]; //заполняем значения  
 price[0] = a1.TicketPrice; price[1] = a2.TicketPrice; price[2] = a3.TicketPrice;  
 price[3] = a4.TicketPrice; price[4] = a5.TicketPrice; price[5] = a6.TicketPrice;  
  
 int[] numbers; //массив с номерами маршрутов  
 numbers = new int[*n*];  
 numbers[0] = a1.RouteNum; numbers[1] = a2.RouteNum; numbers[2] = a3.RouteNum;  
 numbers[3] = a4.RouteNum; numbers[4] = a5.RouteNum; numbers[5] = a6.RouteNum;  
  
 double min = 10000;  
 int num = 0;  
  
 for (int e = 0; e < *n*; e++) { //находим минимальную стоимость  
 if (price[e] < min) {  
 min = price[e];  
 num = numbers[e];  
 }  
 }  
  
 System.*out*.println("The cheapest trip is by bus " + num + ": " + min + " Rub.");  
 }  
  
 static void MostExpensiveTrip(Trip a1, Trip a2, Trip a3, Trip a4, Trip a5, Trip a6) {  
 double[] price; //массив с ценами на поездки  
 price = new double[*n*]; //заполняем значения  
 price[0] = a1.TicketPrice; price[1] = a2.TicketPrice; price[2] = a3.TicketPrice;  
 price[3] = a4.TicketPrice; price[4] = a5.TicketPrice; price[5] = a6.TicketPrice;  
  
 int[] numbers; //массив с номерами маршрутов  
 numbers = new int[*n*];  
 numbers[0] = a1.RouteNum; numbers[1] = a2.RouteNum; numbers[2] = a3.RouteNum;  
 numbers[3] = a4.RouteNum; numbers[4] = a5.RouteNum; numbers[5] = a6.RouteNum;  
  
 double max = 0;  
 int num = 0;  
  
 for (int e = 0; e < *n*; e++) { //находим минимальную стоимость  
 if (price[e] > max) {  
 max = price[e];  
 num = numbers[e];  
 }  
 }  
  
 System.*out*.println("The most expensive trip is by bus " + num + ": " + max + " Rub.");  
 }  
}  
  
class ExpressTrip extends Trip {  
 public double LengthCoefficient;  
 private String StartPoint, FinalPoint; //переменные для назв. нач. и кон. пунктов  
 private int RouteNum; //переменные с номером маршрута  
 private double TravelTime, RouteLen, TicketPrice; //его длительностью, длиной, ценой билета.  
 public ExpressTrip(String SP, String FP, int RN, double TT, double RL, double TP, double LC) {  
 super(SP, FP, RN, TT, RL, TP);  
 this.StartPoint = SP;  
 this.FinalPoint = FP;  
 this.RouteNum = RN;  
 this.TravelTime = TT;  
 this.RouteLen = RL;  
 this.TicketPrice = TP;  
 this.LengthCoefficient = LC;  
 }  
 public ExpressTrip() { //конструктор2  
 this.StartPoint = "Moscow";  
 this.FinalPoint = "Kazan";  
 this.RouteNum = 1015;  
 this.TravelTime = 12.15;  
 this.RouteLen = 820;  
 this.TicketPrice = 1703;  
 this.LengthCoefficient = 0.98;  
 }  
  
 static void ExpressRouteLength (ExpressTrip e1, ExpressTrip e2, ExpressTrip e3,  
 ExpressTrip e4, ExpressTrip e5, ExpressTrip e6) {  
  
 double[] len; //массив с длиной маршрута  
 len = new double[*n*];  
 len[0] = e1.RouteLen; len[1] = e2.RouteLen; len[2] = e3.RouteLen;  
 len[3] = e4.RouteLen; len[4] = e5.RouteLen; len[5] = e6.RouteLen;  
  
 double[] coef; //массив с коэффицентами  
 coef = new double[*n*];  
 coef[0] = e1.LengthCoefficient; coef[1] = e2.LengthCoefficient; coef[2] = e3.LengthCoefficient;  
 coef[3] = e4.LengthCoefficient; coef[4] = e5.LengthCoefficient; coef[5] = e6.LengthCoefficient;  
  
 String[] destination; //массив с пунктами назначения  
 destination = new String[*n*]; // заполняем значения  
 destination[0] = e1.FinalPoint; destination[1] = e2.FinalPoint; destination[2] = e3.FinalPoint;  
 destination[3] = e4.FinalPoint; destination[4] = e5.FinalPoint; destination[5] = e6.FinalPoint;  
  
 for (int l = 0; l < *n*; l++) {  
 double explen = len[l] \* coef[l];  
 System.*out*.println("Express trip to " + destination[l] + " is: " + Math.*ceil*(explen) + " km.");  
 }  
 }  
  
 static void NewPrice(ExpressTrip e1, ExpressTrip e2, ExpressTrip e3,  
 ExpressTrip e4, ExpressTrip e5, ExpressTrip e6) {  
 double[] price; //массив с ценами на поездки  
 price = new double[*n*]; //заполняем значения  
 price[0] = e1.TicketPrice; price[1] = e2.TicketPrice; price[2] = e3.TicketPrice;  
 price[3] = e4.TicketPrice; price[4] = e5.TicketPrice; price[5] = e6.TicketPrice;  
  
 double[] coef; //массив с коэффицентами  
 coef = new double[*n*];  
 coef[0] = e1.LengthCoefficient; coef[1] = e2.LengthCoefficient; coef[2] = e3.LengthCoefficient;  
 coef[3] = e4.LengthCoefficient; coef[4] = e5.LengthCoefficient; coef[5] = e6.LengthCoefficient;  
  
 String[] destination; //массив с пунктами назначения  
 destination = new String[*n*]; // заполняем значения  
 destination[0] = e1.FinalPoint; destination[1] = e2.FinalPoint; destination[2] = e3.FinalPoint;  
 destination[3] = e4.FinalPoint; destination[4] = e5.FinalPoint; destination[5] = e6.FinalPoint;  
  
 for (int l = 0; l < *n*; l++) {  
 double newpr = price[l] \* (1 + (1 - coef[l]));  
 System.*out*.println("A bus ticket to " + destination[l] + " by Express route costs " + Math.*ceil*(newpr) + " Rub.");  
 }  
 }  
}  
  
public class Main {  
  
 public static void main(String[] args) {  
 //заполнение данных об автобусах  
 Trip trip1 = new Trip("Moscow", "Kiev", 560, 10.9, 860, 3405);  
 Trip trip2 = new Trip("Moscow", "Minsk", 915, 7.9, 680, 1589);  
 Trip trip3 = new Trip("Moscow", "Sevastopol", 440, 20.5, 1900, 3105);  
 Trip trip4 = new Trip("Moscow", "Saint-Petersburg", 685, 7.5, 640, 1163);  
 Trip trip5 = new Trip("Moscow", "Arkhangelsk", 590, 16.1, 1200, 2400);  
 Trip trip6 = new Trip();  
  
 //вызов методов(12)  
 Trip.*TimeDifference*(trip1, trip2); //разница во времени между поездками  
 Trip.*ShortestTrip*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*LongestTrip*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*InfoAboutRoutes*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*AverageSpeed*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*LongestRoute*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*ShortestRoute*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*GetPrice*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*StudentSale30*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*ChildTariff50*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*CheapestTrip*(trip1, trip2, trip3, trip4, trip5, trip6);  
 Trip.*MostExpensiveTrip*(trip1, trip2, trip3, trip4, trip5, trip6);  
  
 System.*out*.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  
   
 //данные для нового класса  
 ExpressTrip etrip1 = new ExpressTrip("Moscow", "Kiev", 560, 10.9, 860, 3405, 0.81);  
 ExpressTrip etrip2 = new ExpressTrip("Moscow", "Minsk", 915, 7.9, 680, 1589, 0.76);  
 ExpressTrip etrip3 = new ExpressTrip("Moscow", "Sevastopol", 440, 20.5, 1900, 3105, 0.9);  
 ExpressTrip etrip4 = new ExpressTrip("Moscow", "Saint-Petersburg", 685, 7.5, 640, 1163, 0.85);  
 ExpressTrip etrip5 = new ExpressTrip("Moscow", "Arkhangelsk", 590, 16.1, 1200, 2400, 0.78);  
 // ExpressTrip etrip6 = new ExpressTrip("Moscow", "Kazan", 1015, 12.5, 820, 1703, 0.67);  
 ExpressTrip etrip6 = new ExpressTrip();  
  
 //методы нового класса  
 ExpressTrip.*ExpressRouteLength*(etrip1, etrip2, etrip3, etrip4, etrip5, etrip6);  
 ExpressTrip.*NewPrice*(etrip1, etrip2, etrip3, etrip4, etrip5, etrip6);  
 }  
}

РЕЗУЛЬТАТ

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание