Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

за 1 семестр

По дисциплине: «СПП»

Выполнил:

Студент 3 курса

Группы ПО-5

Лозейко П. А.

Проверил:

Крощенко А. А.

2021

Вариант 9



**Код программы:**

**Program.cs**

using System;

using System.Collections.Generic;

namespace lab4.\_1.\_9

{

class Program

{

static void Main(string[] args)

{

Mobile.Phone.Camera LowCamera = new Mobile.Phone.Camera(1, "no name");

Mobile.Phone.Camera MediumCamera = new Mobile.Phone.Camera(32, "Optimus 3D");

Mobile.Phone.Camera HighCamera = new Mobile.Phone.Camera(108, "808 PureView");

Mobile.Phone IPhone8 = new Mobile.Phone("IPhone 8");

Mobile.Phone Nokia3310 = new Mobile.Phone("Nokia 3310");

Mobile.Phone OnePlus8Pro = new Mobile.Phone("OnePlus 8 Pro");

IPhone8.AddCamera(MediumCamera);

IPhone8.AddCamera(HighCamera);

Nokia3310.AddCamera(LowCamera);

OnePlus8Pro.AddCamera(HighCamera);

OnePlus8Pro.AddCamera(HighCamera);

OnePlus8Pro.AddCamera(MediumCamera);

List<Mobile.Phone> Mobiles1 = new List<Mobile.Phone>();

Mobiles1.Add(IPhone8);

Mobiles1.Add(Nokia3310);

Mobiles1.Add(OnePlus8Pro);

Mobile mobiles = new Mobile(new string("Mobiles 1"));

mobiles.SetPositions(Mobiles1);

Console.WriteLine(mobiles.GetName() + ":");

foreach (Mobile.Phone phone in mobiles.GetPositions())

{

Console.WriteLine("\t" + phone.GetName() + ":");

foreach (Mobile.Phone.Camera camera in phone.GetCamera())

{

Console.WriteLine("\t\t" + camera.GetName() + "\t" + camera.GetMegapixels() + " Megapixels");

}

Console.WriteLine();

}

}

}

}

**Mobile.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab4.\_1.\_9

{

public class Mobile

{

private string Name;

private List<Phone> Cameras = new List<Phone>();

public Mobile()

{

}

public Mobile(string name)

{

Name = name;

}

public Mobile(string name, List<Phone> cameras)

{

Name = name;

Cameras = cameras;

}

public void SetName(string name)

{

Name = name;

}

public string GetName()

{

return Name;

}

public void SetPositions(List<Phone> cameras)

{

Cameras = cameras;

}

public List<Phone> GetPositions()

{

return Cameras;

}

public void AddPosition(Phone phone)

{

Cameras.Add(phone);

}

public void RemovePosition(Phone phone)

{

Cameras.Remove(phone);

}

public class Phone

{

private string Name;

private string CPU;

private int Memory;

private List<Camera> Cameras = new List<Camera>();

public Phone()

{

}

public Phone(string name)

{

Name = name;

}

public Phone(string name, string cpu)

{

Name = name;

CPU = cpu;

}

public Phone(string name, string cpu, int memory)

{

Name = name;

CPU = cpu;

Memory = memory;

}

public Phone(string name, string cpu, int memory, List<Camera> camera)

{

Name = name;

CPU = cpu;

Memory = memory;

}

public Phone(string name, List<Camera> cameras)

{

Name = name;

Cameras = cameras;

}

public void SetName(string name)

{

Name = name;

}

public string GetName()

{

return Name;

}

public void SetCPU(string cpu)

{

CPU = cpu;

}

public string GetCPU()

{

return CPU;

}

public void SetMemory(int memory)

{

Memory = memory;

}

public int GetMemory()

{

return Memory;

}

public void SetCamera(List<Camera> cameras)

{

Cameras = cameras;

}

public List<Camera> GetCamera()

{

return Cameras;

}

public void AddCamera(Camera camera)

{

Cameras.Add(camera);

}

public void RemoveCamera(Camera camera)

{

Cameras.Remove(camera);

}

public class Camera

{

private int Megapixels;

private string Name;

public Camera(int megapixels, string name)

{

Megapixels = megapixels;

Name = name;

}

public void SetMegapixels(int megapixels)

{

Megapixels = megapixels;

}

public int GetMegapixels()

{

return Megapixels;

}

public void SetName(string name)

{

Name = name;

}

public string GetName()

{

return Name;

}

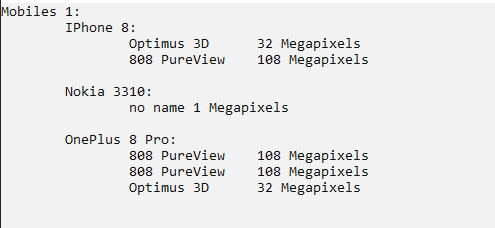
}

}

}

}

**Результат работы программы:**

****

****

**Код программы:**

**Program.cs**

using System;

namespace lab4.\_2.\_9

{

class Program

{

static void Main(string[] args)

{

Wheel Continental = new Wheel(15, 220);

Car Porsche = new Car(Continental);

Porsche.Print();

}

}

}

**Car.cs**

using System;

using System.Collections.Generic;

namespace lab4.\_2.\_9

{

class Wheel

{

public int radius;

public int width;

public Wheel(int r, int w)

{

radius = r;

width = w;

}

}

class Car

{

string model = "Porsche Panamera";

Wheel wheel;

public Car(Wheel someWheel)

{

wheel = someWheel;

}

public void Print()

{

Console.WriteLine($"Модель: {model}. Колёса: радиус - {wheel.radius}, ширина - {wheel.width}");

}

public void ChangeWheel(int r, int w)

{

wheel.radius = r;

wheel.width = w;

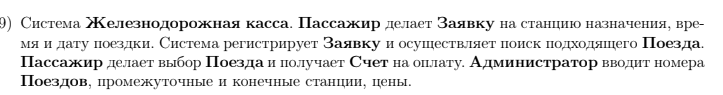
}

}

}

**Результат работы программы:**

****

****

**Administrator.cs**

using System;

using System.Collections.Generic;

namespace lab4.\_3.\_9

{

public class Administrator : Person

{

private RailwayTicketOffice railwayTicketOffice = new RailwayTicketOffice();

public Administrator(int age, string name, RailwayTicketOffice railwayTicketOffice) : base(age, name)

{

this.railwayTicketOffice = railwayTicketOffice;

}

public void SetRailwayTicketOffice(RailwayTicketOffice railwayTicketOffice)

{

this.railwayTicketOffice = railwayTicketOffice;

}

public RailwayTicketOffice GetRailwayTicketOffice()

{

return railwayTicketOffice;

}

public void AddOrder(Order order)

{

railwayTicketOffice.AddOrder(order);

}

}

}

**Check.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab4.\_3.\_9

{

public class Check

{

Check()

{

}

}

}

**Order.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab4.\_3.\_9

{

public class Order

{

private Train Train = null;

private Passenger Passenger = null;

public Order(Train train, Passenger passenger)

{

Train = train;

Passenger = passenger;

}

public void SetTrain(Train train)

{

Train = train;

}

public Train GetTrain()

{

return Train;

}

public void SetPassenger(Passenger passenger)

{

Passenger = passenger;

}

public Passenger GetPassenger()

{

return Passenger;

}

}

}

**Passenger.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab4.\_3.\_9

{

public class Passenger : Person

{

private List<Order> Orders = new List<Order>();

private int Bill = 0;

private Check Check = null;

public Passenger(int Age, string Name) : base(Age, Name)

{

}

public Passenger(int Age, string Name, int Bill) : base(Age, Name)

{

this.Bill = Bill;

}

public Passenger(int Age, string Name, int Bill, List<Order> Orders) : base(Age, Name)

{

this.Bill = Bill;

this.Orders = Orders;

}

public Passenger(int Age, string Name, int Bill, List<Order> Orders, Check Check) : base(Age, Name)

{

this.Bill = Bill;

this.Orders = Orders;

this.Check = Check;

}

public void SetOrders(List<Order> orders)

{

Orders = orders;

}

public List<Order> GetOrders()

{

return Orders;

}

public void SetCheck(Check check)

{

Check = check;

}

public Check GetCheck()

{

return Check;

}

public void SetBill(int bill)

{

Bill = bill;

}

public int GetBill()

{

return Bill;

}

public void AddOrder(Order order)

{

Orders.Add(order);

}

public void RemoveOrder(Order order)

{

Orders.Remove(order);

}

public bool AddCheck(Train train)

{

if(Bill - train.GetPrice() >= 0)

{

Bill -= train.GetPrice();

return true;

}

return false;

}

public bool RequestOrderAndCheck(RailwayTicketOffice railwayTicketOffice, string StartStation, string EndStation, TrainStation trainStation)

{

var temp = railwayTicketOffice.TrainSearch(StartStation, EndStation);

if (temp.Item1 != -1)

{

Order order = railwayTicketOffice.ProcessOrder(this, temp.Item2);

if (order == null)

{

return false;

}

AddOrder(order);

return AddCheck(temp.Item2);

}

return false;

}

}

}

**Person.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab4.\_3.\_9

{

public class Person

{

private int Age = 0;

private string Name;

public Person(int age, string name)

{

Age = age;

Name = name;

}

public void SetAge(int age)

{

Age = age;

}

public int GetAge()

{

return Age;

}

public void SetName(string name)

{

Name = name;

}

public string GetName()

{

return Name;

}

}

}

**Program.cs**

using System;

using System.Collections.Generic;

namespace lab4.\_3.\_9

{

class Program

{

static List<string> RandomStatoins(int NumStations)

{

List<string> stations = new List<string>();

if (NumStations <= 0)

{

return stations;

}

string[] stationsName = { "Брест", "Минск", "Витебск", "Гомель", "Гродно", "Могилёв", "Бобруйск", "Барановичи", "Новополоцк", "Пинск", "Борисов", "Лида", "Мозырь", "Полоцк", "Слоним", "Орша", "Молодечно", "Жлобин", "Кобрин", "Слуцк" };

Random rand = new Random();

for(int i = 0; i < NumStations; i++)

{

stations.Add(stationsName[rand.Next(0, stationsName.Length)]);

}

return stations;

}

public static DateTime RandomDateTime(DateTime min, DateTime max)

{

Random \_ran = new Random();

return DateTime.MinValue.Add(TimeSpan.FromTicks(min.Ticks + (long)(\_ran.NextDouble() \* (max.Ticks - min.Ticks))));

}

static List<DateTime> GenerateTime(int NumStations)

{

List<DateTime> dateTimes = new List<DateTime>();

DateTime maxdate = new DateTime(2021, 12, 10, 0, 0, 0);

for(int i = 0; i < NumStations; i++)

{

dateTimes.Add(RandomDateTime(DateTime.Now, maxdate));

}

return dateTimes;

}

static void Main(string[] args)

{

int NumStations = 10;

TrainStation trainStation = new TrainStation();

trainStation.AddTrain(new Train(RandomStatoins(NumStations), GenerateTime(NumStations), 10, 1));

trainStation.AddTrain(new Train(RandomStatoins(NumStations), GenerateTime(NumStations), 15, 2));

trainStation.AddTrain(new Train(RandomStatoins(NumStations), GenerateTime(NumStations), 12, 3));

trainStation.AddTrain(new Train(RandomStatoins(NumStations), GenerateTime(NumStations), 11, 4));

trainStation.AddTrain(new Train(RandomStatoins(NumStations), GenerateTime(NumStations), 13, 5));

RailwayTicketOffice railwayTicketOffice = new RailwayTicketOffice();

railwayTicketOffice.SetTrainStation(trainStation);

Administrator Administrator = new Administrator(38, "Kirill Smekalov", railwayTicketOffice);

Passenger Passenger = new Passenger(19, "Stas Sokolov", 100);

railwayTicketOffice.AddAdministrator(Administrator);

railwayTicketOffice.AddPassenger(Passenger);

if (Passenger.RequestOrderAndCheck(railwayTicketOffice, "Брест", "Минск", trainStation))

{

Console.WriteLine("Train successfully found");

}

else

{

Console.WriteLine("Train not found");

}

}

}

}

**RailwayTicketoffice.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab4.\_3.\_9

{

public class RailwayTicketOffice

{

private TrainStation TrainStations = new TrainStation();

private List<Train> Trains = new List<Train>();

private List<Administrator> Administrators = new List<Administrator>();

private List<Passenger> Passengers = new List<Passenger>();

private List<Order> Orders = new List<Order>();

public RailwayTicketOffice()

{

}

public RailwayTicketOffice(TrainStation trainStations)

{

TrainStations = trainStations;

}

public RailwayTicketOffice(TrainStation trainStations, List<Train> trains)

{

TrainStations = trainStations;

Trains = trains;

}

public RailwayTicketOffice(TrainStation trainStations, List<Train> trains, List<Administrator> administrators)

{

TrainStations = trainStations;

Trains = trains;

Administrators = administrators;

}

public RailwayTicketOffice(TrainStation trainStations, List<Train> trains, List<Administrator> administrators, List<Passenger> passengers)

{

TrainStations = trainStations;

Trains = trains;

Administrators = administrators;

Passengers = passengers;

}

public RailwayTicketOffice(TrainStation trainStations, List<Train> trains, List<Administrator> administrators, List<Passenger> passengers, List<Order> orders)

{

TrainStations = trainStations;

Trains = trains;

Administrators = administrators;

Passengers = passengers;

Orders = orders;

}

public void SetTrainStation(TrainStation trainStations)

{

TrainStations = trainStations;

}

public TrainStation GetCatalog()

{

return TrainStations;

}

public void SetTrain(List<Train> train)

{

Trains = train;

}

public List<Train> GetTrain()

{

return Trains;

}

public void SetAdministrator(List<Administrator> administrator)

{

Administrators = administrator;

}

public List<Administrator> GetAdministrator()

{

return Administrators;

}

public void SetPassengers(List<Passenger> passenger)

{

Passengers = passenger;

}

public List<Passenger> GetPassenger()

{

return Passengers;

}

public void SetOrder(List<Order> order)

{

Orders = order;

}

public List<Order> GetOrder()

{

return Orders;

}

public void AddTrain(Train train)

{

TrainStations.AddTrain(train);

}

public void RemoveTrain(Train train)

{

TrainStations.RemoveTrain(train);

}

public void AddAdministrator(Administrator administrator)

{

Administrators.Add(administrator);

}

public void RemoveAdministrator(Administrator administrator)

{

Administrators.Remove(administrator);

}

public void AddPassenger(Passenger passenger)

{

Passengers.Add(passenger);

}

public void RemovePassenger(Passenger passenger)

{

Passengers.Remove(passenger);

}

public void AddOrder(Order order)

{

Orders.Add(order);

}

public void RemoveOrder(Order order)

{

Orders.Remove(order);

}

public Tuple<int, Train> TrainSearch(string StartStation, string EndStation)

{

return TrainStations.TrainSearch(StartStation, EndStation);

}

public Order ProcessOrder(Passenger passenger, Train train)

{

Train orderedTrain = TrainStations.GiveTrain(train);

if (orderedTrain == null)

{

return null;

}

Order order = new Order(orderedTrain, passenger);

bool processed = false;

while (!processed)

{

foreach (Administrator administrator in Administrators)

{

administrator.AddOrder(order);

processed = true;

break;

}

}

Orders.Add(order);

return order;

}

}

}

**Train.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

namespace lab4.\_3.\_9

{

public class Train

{

private List<string> Stations = new List<string>();

private List<DateTime> TrainTime = new List<DateTime>();

private int Price = 1;

private int ID = 0;

public Train(List<string> stations, List<DateTime> trainTime, int price, int id)

{

Stations = stations;

TrainTime = trainTime;

Price = price;

ID = id;

}

public Train(List<string> stations, List<DateTime> trainTime)

{

Stations = stations;

TrainTime = trainTime;

}

public void SetStations(List<string> stations)

{

Stations = stations;

}

public List<string> GetStations()

{

return Stations;

}

public void SetTrainTime(List<DateTime> trainTime)

{

TrainTime = trainTime;

}

public List<DateTime> GetTrainTime()

{

return TrainTime;

}

public void SetPrice(int price)

{

Price = price;

}

public int GetPrice()

{

return Price;

}

public void SetID(int id)

{

ID = id;

}

public int GetID()

{

return ID;

}

public bool \_Equals(object Other)

{

if (GetStations().Equals(((Train)Other).GetStations()) && GetTrainTime().Equals(((Train)Other).GetTrainTime()) && GetPrice().Equals(((Train)Other).GetPrice()) && GetID().Equals(((Train)Other).GetID()))

{

return true;

}

return false;

}

}

}

**TrainStation.cs**

using System;

using System.Collections.Generic;

using System.Text;

namespace lab4.\_3.\_9

{

public class TrainStation

{

private List<Train> Trains = new List<Train>();

public TrainStation()

{

}

public TrainStation(List<Train> train)

{

Trains = train;

}

public void SetTrains(List<Train> train)

{

Trains = train;

}

public List<Train> GetTrains()

{

return Trains;

}

public void AddTrain(Train train)

{

Trains.Add(train);

}

public void RemoveTrain(Train train)

{

Trains.Remove(train);

}

public Tuple<int, Train> TrainSearch(string StartStation, string EndStation)

{

List<string> temp;

for (int i = 0; i < Trains.Count; i++)

{

temp = Trains[i].GetStations();

if (temp.IndexOf(StartStation) < temp.IndexOf(EndStation) && temp.IndexOf(StartStation) != -1)

{

return Tuple.Create(i, Trains[i]);

}

}

Train train = null;

return Tuple.Create(-1, train);

}

public Train GiveTrain(Train train)

{

int index = -1;

for (int i = 0; i < Trains.Count; i++)

{

if (Trains[i].GetStations() == train.GetStations() && Trains[i].GetTrainTime() == train.GetTrainTime() && Trains[i].GetPrice() == train.GetPrice() && Trains[i].GetID() == train.GetID())

{

index = i;

}

}

if (index == -1)

{

return null;

}

Train result = Trains[index];

RemoveTrain(train);

return result;

}

}

}

**Результат работы программы:**

****