# Лабораторная работа № 15

**Тема:** Разработка и тестирование программ на языке С# с использованием принципа инкапсуляции.

**Цель работы:**

1. Формирование навыков разработки программ на языке С# с использованием принципа инкапсуляции.
2. Изучение способов описания и использования классов на языке С#.

. **Оборудование:**

1. ПК
2. Программное обеспечение: ОС Windows, среда Visual Studio.Net

**Контрольные вопросы:**

1. Что такое инкапсуляция?
2. Что такое свойства и для чего их используют?
3. Записать стандартное определение свойства и объяснить?
4. В каких случаях используют автоматические свойства?
5. Как создать конструктор класса?

**Ход работы:**

1. Разработать алгоритм задачи и представить его в виде  программы на С#
2. Используя средства Visual Studio.Net разработать программу и  выполнить ее тестирование и отладку
3. Результаты представить в виде отчета
4. Сделать вывод о проделанной работе

**Варианты заданий для лабораторной работы №15**

Разработать классы для описанных ниже объектов. Включить в класс set, get Определить методы. Разработать приложение демонстрирующее работу с этими методами

Разработать класс для объекта Train: Пункт назначения, Номер поезда, Время отправления, Число общих мест, Купейных, Плацкартных. Создать массив объектов. Вывести: а) список поездов, следующих до заданного пункта назначения; б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа; в) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

**Листинг с исходным кодом**

using System;

using System.Net;

class Train

{

private string destinat; // Пункт назначения

private int trainNumber; // Номер поезда

private string departureTime; // Время отправления

private int totalSeats; // Число общих мест

private int coupeSeats; // Число купейных мест

private int reservedSeats; // Число плацкартных мест

public string Destinat

{

get { return destinat; }

set { destinat = value; }

}

public int TrainNumber

{

get { return trainNumber; }

set { trainNumber = value; }

}

public string DepartureTime

{

get { return departureTime; }

set { departureTime = value; }

}

public int TotalSeats

{

get { return totalSeats; }

set { totalSeats = value; }

}

public int CoupeSeats

{

get { return coupeSeats; }

set { coupeSeats = value; }

}

public int ReservedSeats

{

get { return reservedSeats; }

set { reservedSeats = value; }

}

public void Print()

{

Console.WriteLine($"Пункт назначения:{Destinat} \t Номер поезда: {TrainNumber}\t Время отправления: {DepartureTime}\t Число общих мест: {TotalSeats}\t Число купейных мест: {CoupeSeats}\t Число плацкартных мест: {ReservedSeats}");

}

public Train(string destinat, int trainNumber, string departureTime, int totalSeats, int coupeSeats, int reservedSeats)

{

Destinat = destinat;

TrainNumber = trainNumber;

DepartureTime = departureTime;

TotalSeats = totalSeats;

CoupeSeats = coupeSeats;

ReservedSeats = reservedSeats;

}

~Train()

{

Console.WriteLine($"Поезд с номером {TrainNumber} удалён.");

}

}

class Program

{

static void Main()

{

Console.WriteLine("Введите количество поездов: ");

int count = Convert.ToInt32(Console.ReadLine());

Train[] trains = new Train[count];

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

{

Console.WriteLine("Введите данные для поезда {0}: ", i + 1);

Console.Write("Пункт назначения: ");

string destinat = Convert.ToString(Console.ReadLine());

Console.Write("Номер поезда: ");

int trainNumber = Convert.ToInt32(Console.ReadLine());

Console.Write("Время отправления : ");

string departureTime = Console.ReadLine();

Console.Write("Число общих мест: ");

int totalSeats = Convert.ToInt32(Console.ReadLine());

Console.Write("Число купейных мест: ");

int coupeSeats = Convert.ToInt32(Console.ReadLine());

Console.Write("Число плацкартных мест: ");

int reservedSeats = Convert.ToInt32(Console.ReadLine());

trains[i] = new Train(destinat, trainNumber, departureTime, totalSeats, coupeSeats, reservedSeats);

}

Console.WriteLine("\nСписок поездов до заданного пункта назначения: ");

string destant = Console.ReadLine();

Console.WriteLine("а)");

for (int i = 0; i < trains.Length; i++)

{

if (trains[i].Destinat == destant) trains[i].Print();

}

Console.WriteLine("\nСписок поездов до заданного пункта назначения и отправляющихся после заданного часа: ");

string departTime = Console.ReadLine();

Console.WriteLine("б)");

for (int i = 0; i < trains.Length; i++)

{

if (trains[i].Destinat == destant && trains[i].DepartureTime == departTime) trains[i].Print();

}

Console.WriteLine("\nСписок поездов, отправляющихся до заданного пункта назначения и имеющих общие места: ");

int totSeats = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("в)");

for (int i = 0; i < trains.Length; i++)

{

if (trains[i].Destinat == destant && trains[i].TotalSeats == totSeats) trains[i].Print();

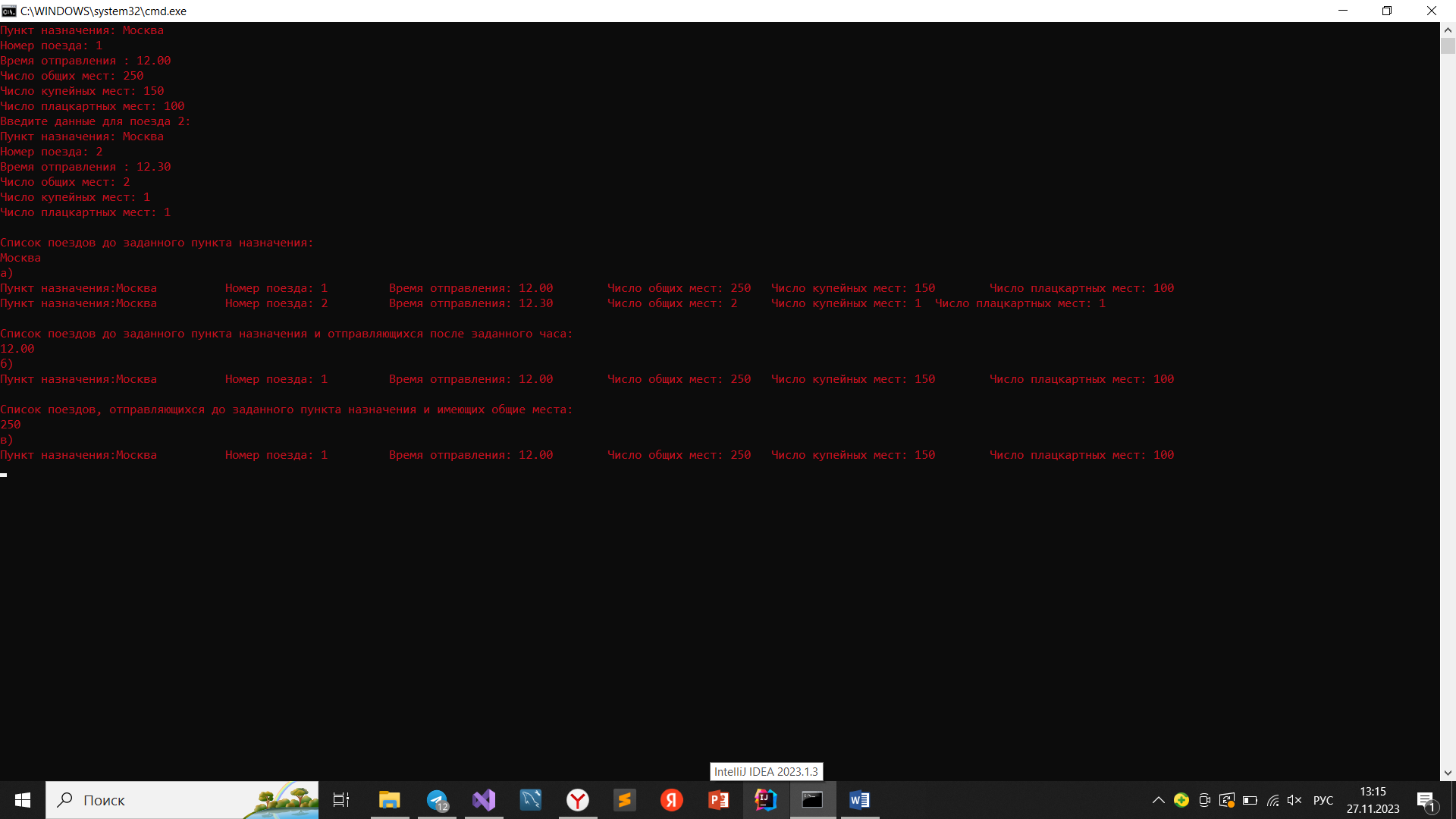
}

Console.ReadKey();

}

}

**Результаты тестирования**



Вывод по работе.