# Лабораторная работа № 14

**Тема:** Разработка и тестирование программ на языке С# с использованием конструкторов и деструкторов классов.

**Цель работы:**

1. формирование навыков разработки программ на языке С# с использованием конструкторов и деструкторов классов;
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 конструктор?

Конструктор - это специальный метод класса, который вызывается при создании нового объекта этого класса. Он используется для инициализации полей объекта и выполнения других необходимых операций.

1. Для чего используют конструктор?

Конструктор используется для установки начальных значений полей объекта и выполнения других инициализационных действий, чтобы объект был валидным и готовым к использованию.

1. Что такое конструктор по умолчанию и как он создается?

Конструктор по умолчанию - это конструктор без параметров, который автоматически создается компилятором, если не определено ни одного другого конструктора в классе. Для его создания не требуется никаких дополнительных действий, он уже присутствует в классе, если не определено иного.

1. Для чего используется оператор **new**?

Оператор new используется для создания экземпляра (объекта) класса, выделяя для него память и вызывая конструктор класса.

1. Как создать конструктор класса?

Конструктор класса можно создать, объявив метод с названием класса, который не возвращает значение и не имеет возвращаемого типа (не декларирует void).

1. Что такое деструктор и для чего он используется?

Деструктор - это метод класса, который вызывается автоматически при уничтожении объекта (когда объект больше не используется). Деструктор используется для выполнения операций очистки или освобождения ресурсов, выделенных объектом (например, закрытие файлов или соединений с базой данных).

1. Синтаксис деструктора.

Синтаксис деструктора в C# следующий:

~ClassName()

{

// код деструктора

}

1. Могут ли быть параметры у деструктора?

Деструктор в C# не может принимать параметров и не имеет модификатора доступа (например, public или private). Деструктор вызывается по умолчанию без передачи аргументов.

**Ход работы:**

1. Разработать алгоритм задачи и представить его в виде схемы программы
2. Используя средства среды Visual Studio.Net создать файл с программой и выполнить тестирование и отладку.
3. Результаты представить в виде отчета
4. Сделать вывод о проделанной работе

**Листинг с исходным кодом**

using System;

using System.Net;

class Train

{

public string Destinat; // Пункт назначения

public int TrainNumber; // Номер поезда

public string DepartureTime; // Время отправления

public int TotalSeats; // Число общих мест

public int CoupeSeats; // Число купейных мест

public int ReservedSeats; // Число плацкартных мест

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();

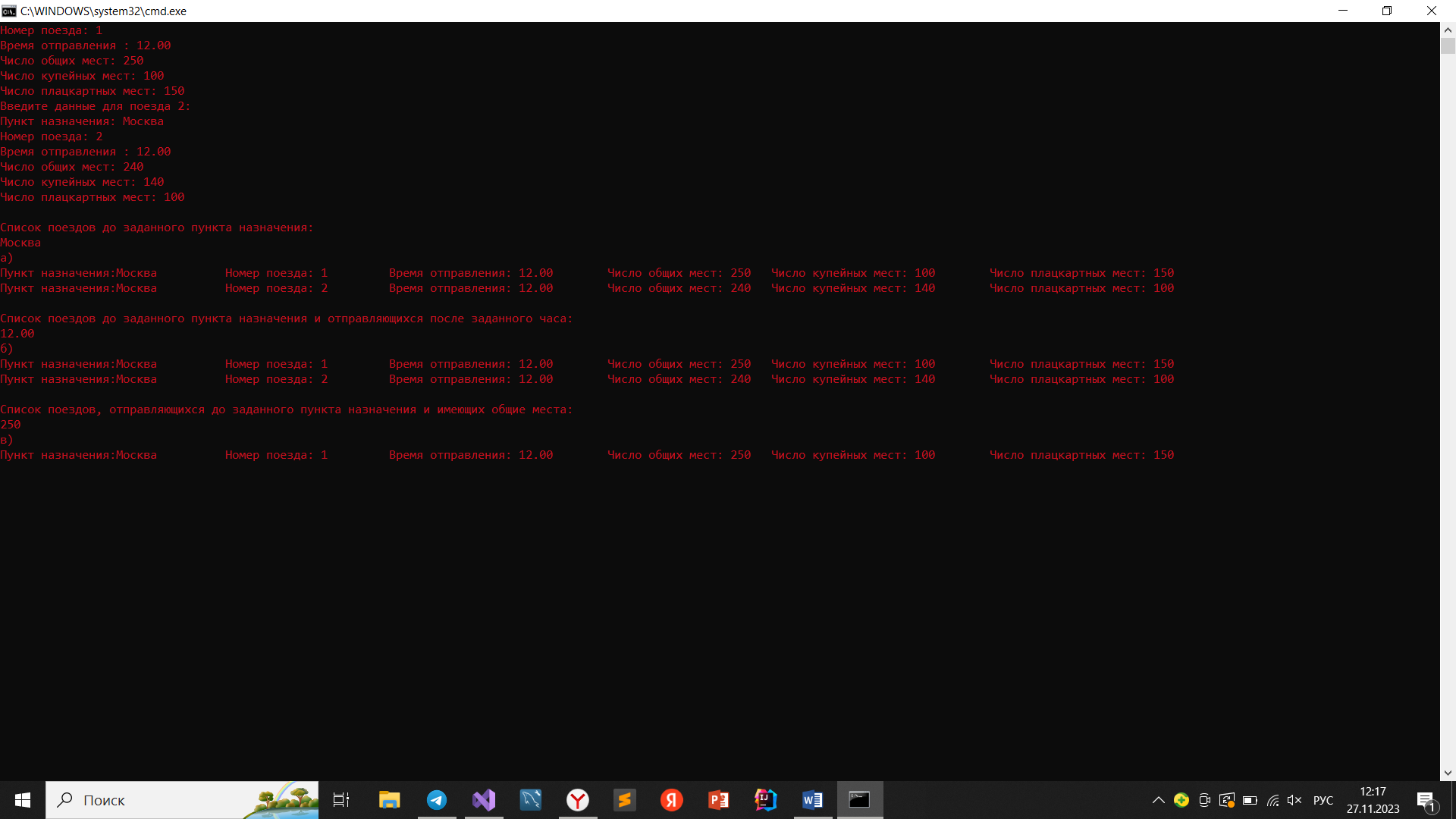
}

Console.ReadKey();

}

}

**Результаты тестирования**



**Используя варианты заданий к лабораторной работе №12-13 разработать классы с использованием конструкторов (один или несколько). Предусмотреть 1 или несколько деструкторов**

**Варианты заданий**

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.

**Вывод по работе.**

Конструкторы используются для инициализации объектов класса при их создании. Они определяются внутри класса и могут принимать параметры. Конструктор по умолчанию создается автоматически, если не определены другие конструкторы.

Деструкторы используются для выполнения операций очистки или освобождения ресурсов, когда объект больше не используется. Они определяются внутри класса с использованием синтаксиса "~ClassName()".