Геронок Артём ИПО-22.24

Основные понятия объектно-ориентированного программирования.

1. Инкапсуляция

1.1

```
  csharp

using System;


public class Student

{

    public string name;

}


class Program

{

    static void Main()

    {

        Student student = new Student();

        student.name = "Иван";

        Console.WriteLine($"Имя студента: {student.name}");

    }

}
```

1.2

```
using System;

public class Car
{
    public int year;
}

class Program
{
    static void Main()
    {
        Car car = new Car();
        car.year = 2020;
        Console.WriteLine($"Год выпуска автомобиля: {car.year}");
    }
}
```

1.3

```
using System;

public class Point
{
    public int x;
}

class Program
{
    static void Main()
    {
        Point point = new Point();
        point.x = 10;
        Console.WriteLine($"Координата X: {point.x}");
    }
}
```

1.4

```
using System;

public class Person
{
    public int age;

    public void Print()
    {
        Console.WriteLine($"Возраст: {age}");
    }
}

class Program
{
    static void Main()
    {
        Person person = new Person();
        person.age = 25;
        person.Print();
    }
}
```

1.5

```
using System;

public class Table
{
    public int rows;
    public int cols;

    public void Display()
    {
        Console.WriteLine($"Строки: {rows}, Столбцы: {cols}");
    }
}

class Program
{
    static void Main()
    {
        Table table = new Table();
}

        table.rows = 5;

        table.cols = 3;

        table.Display();

    }

}
```

1.6

```
using System;

public class Manager
{
    public int age;
    public string name;

    public int GetAge()
    {
        return age;
    }

    public string GetName()
    {
        return name;
    }
}

class Program
```

```
{
    static void Main()
    {
        Manager manager = new Manager();
        manager.age = 35;
        manager.name = "Алексей";

        Console.WriteLine($"Имя: {manager.GetName()}, Возраст:
{manager.GetAge()}");
    }
}
```

1.7

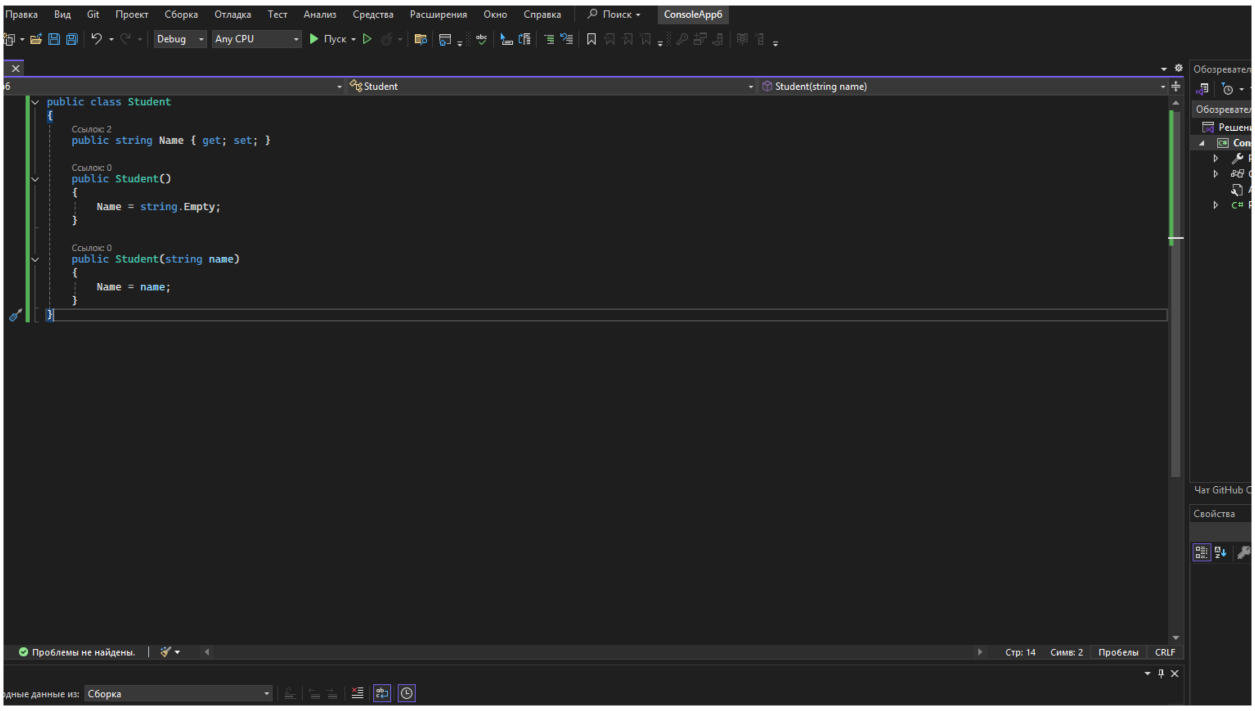
```
using System;

public class Point3D
{
    public int x;
    public int y;
    public int z;

    public void Show()
    {
        Console.WriteLine($"X: {x}, Y: {y}, Z: {z}");
    }
}

class Program
{
    static void Main()
    {
        Point3D point = new Point3D();
        point.x = 5;
}

        point.y = 10;
        point.z = 15;
        point.Show();
    }
}
```

1.8

```
using System;

public class Shop
{
    public string name;

    public string GetName()
    {
        return name;
    }

    public void SetName(string newName)
    {
        name = newName;
    }
}

class Program
```

```
{
    static void Main()
    {
        Shop shop = new Shop();
        shop.SetName("Продукты");
        Console.WriteLine($"Название магазина: {shop.GetName()}");
    }
}
```

2.1

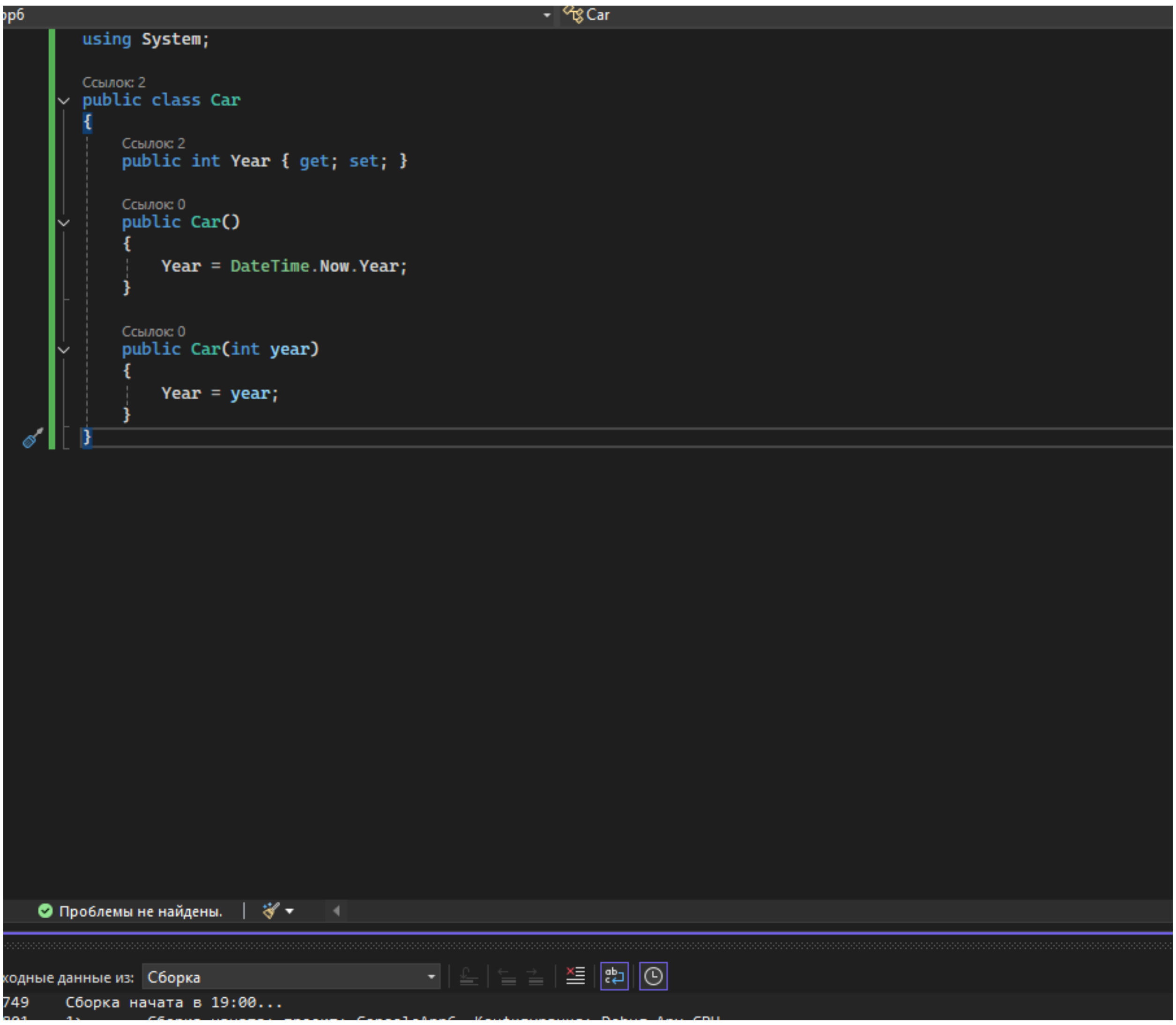


```
public class Student
{
    Ссылок: 2
    public string Name { get; set; }

    Ссылок: 0
    public Student()
    {
        Name = string.Empty;
    }

    Ссылок: 0
    public Student(string name)
    {
        Name = name;
    }
}
```

Проблемы не найдены.

Стр: 14   Симв: 2   Пробелы   CRLF

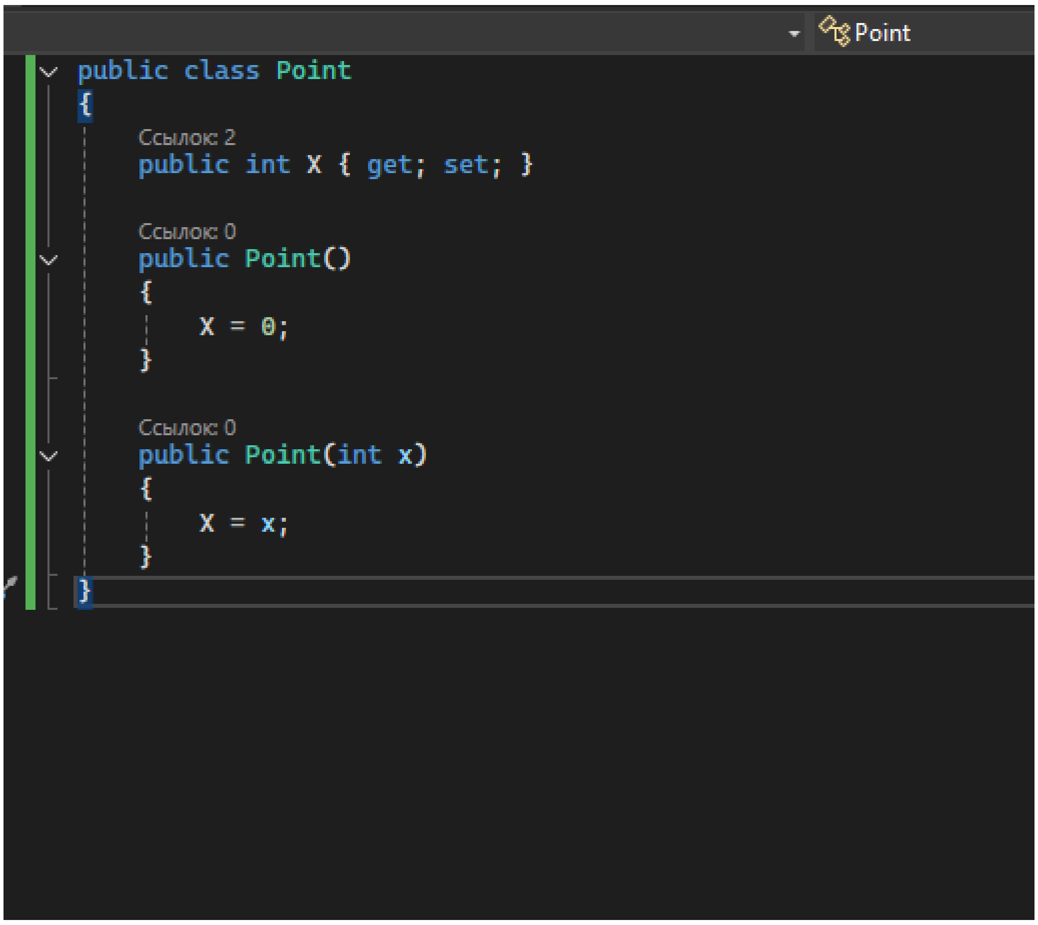
2.2



```
using System;

Ссылок: 2
public class Car
{
    Ссылок: 2
    public int Year { get; set; }

    Ссылок: 0
    public Car()
    {
        Year = DateTime.Now.Year;
    }

    Ссылок: 0
    public Car(int year)
    {
        Year = year;
    }
}
```

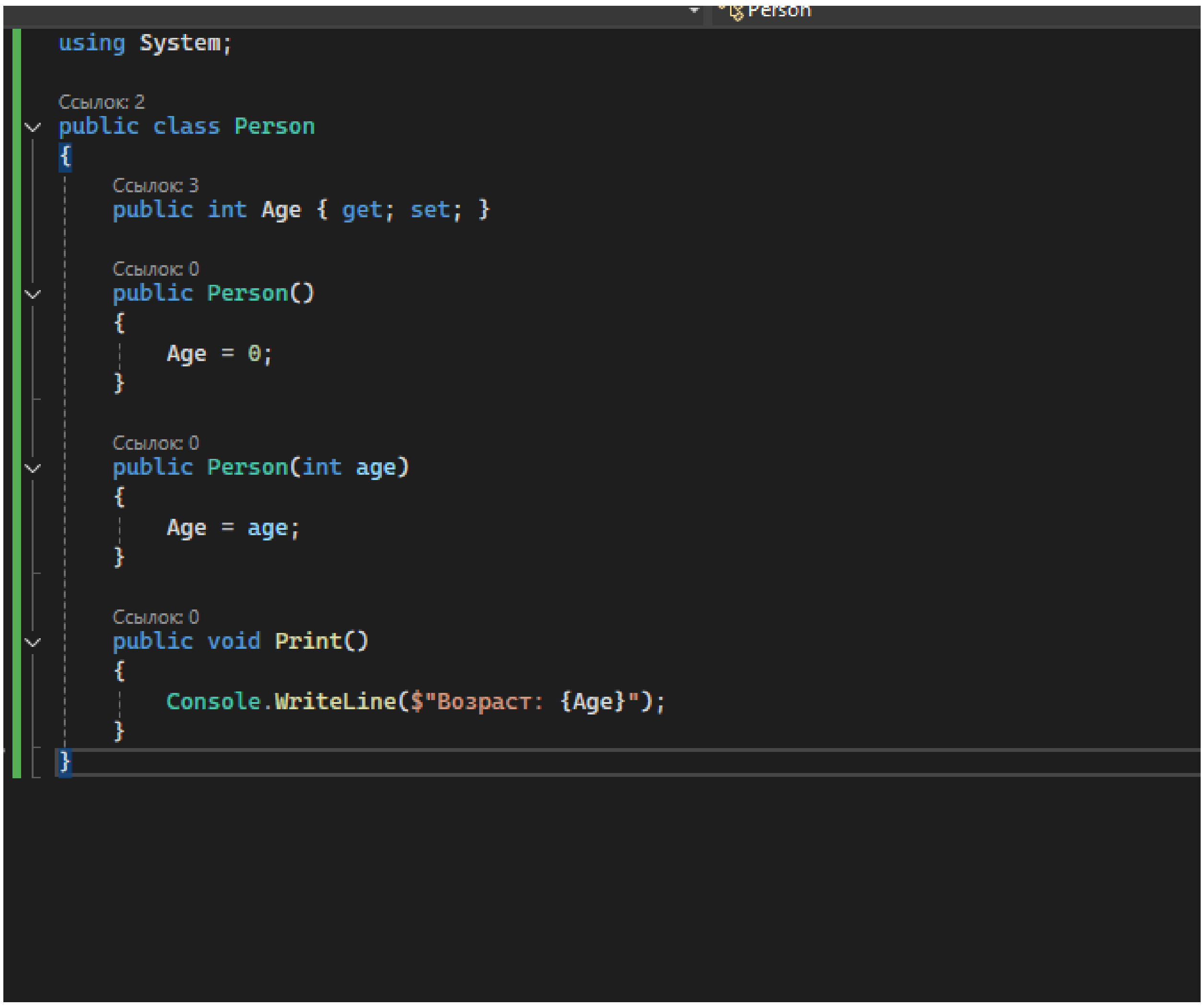
```
749    Сборка начата в 19:00...
```

2.3



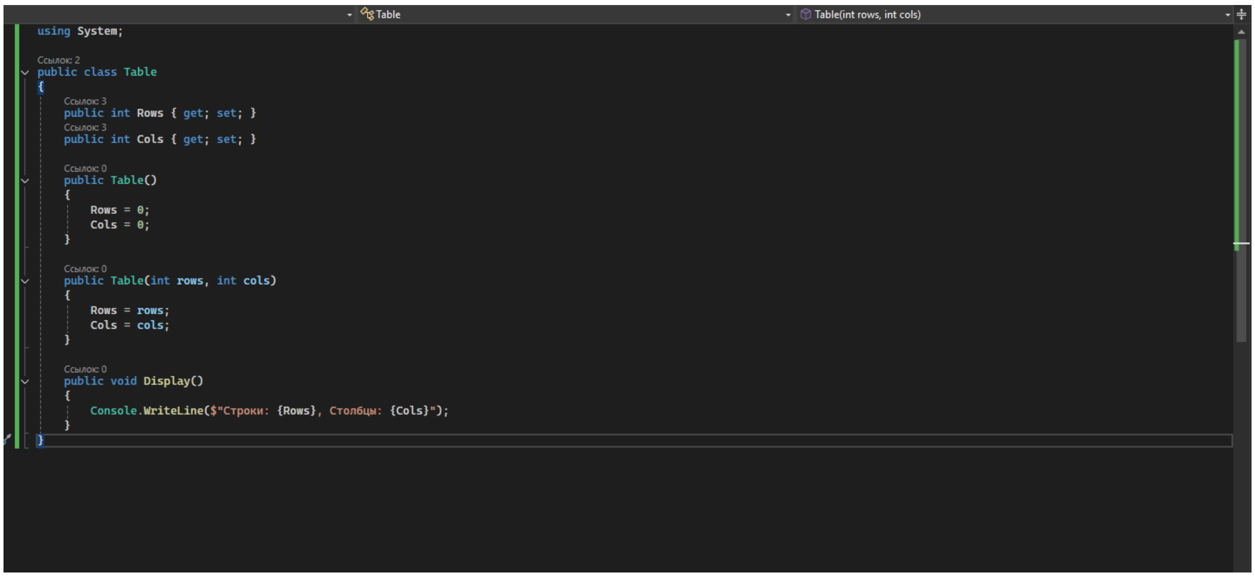
```
public class Point
{
    public int X { get; set; }

    public Point()
    {
        X = 0;
    }

    public Point(int x)
    {
        X = x;
    }
}
```

2.4



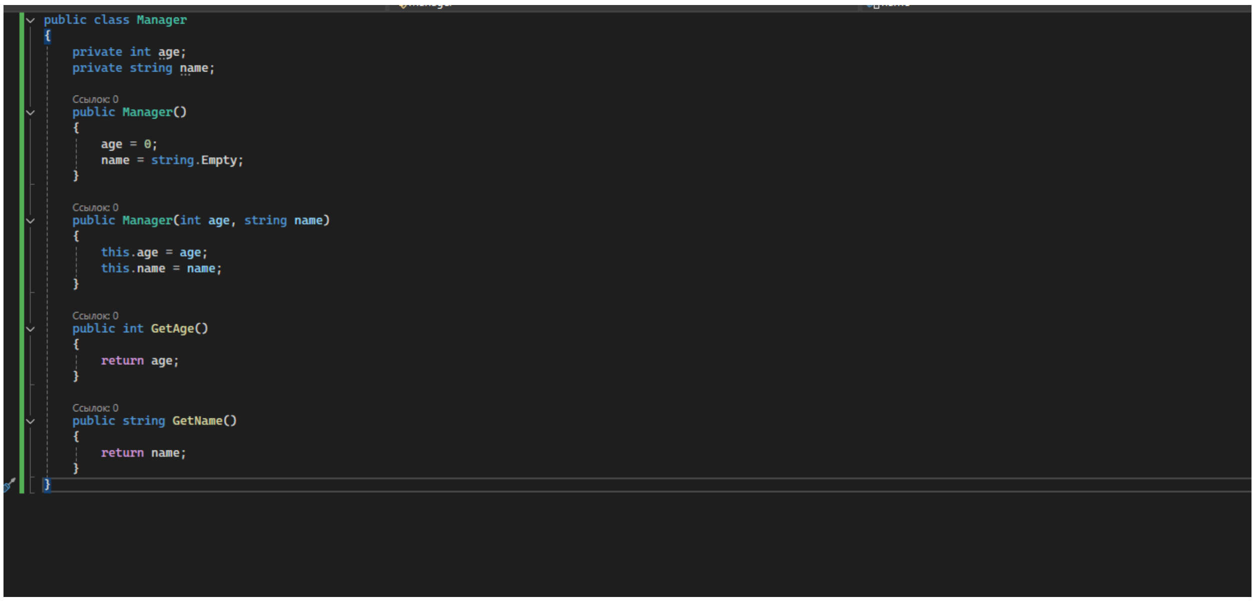
```
using System;

Ссылок: 2
public class Person
{
    Ссылок: 3
    public int Age { get; set; }

    Ссылок: 0
    public Person()
    {
        Age = 0;
    }

    Ссылок: 0
    public Person(int age)
    {
        Age = age;
    }

    Ссылок: 0
    public void Print()
    {
        Console.WriteLine($"Возраст: {Age}");
    }
}
```

2.5



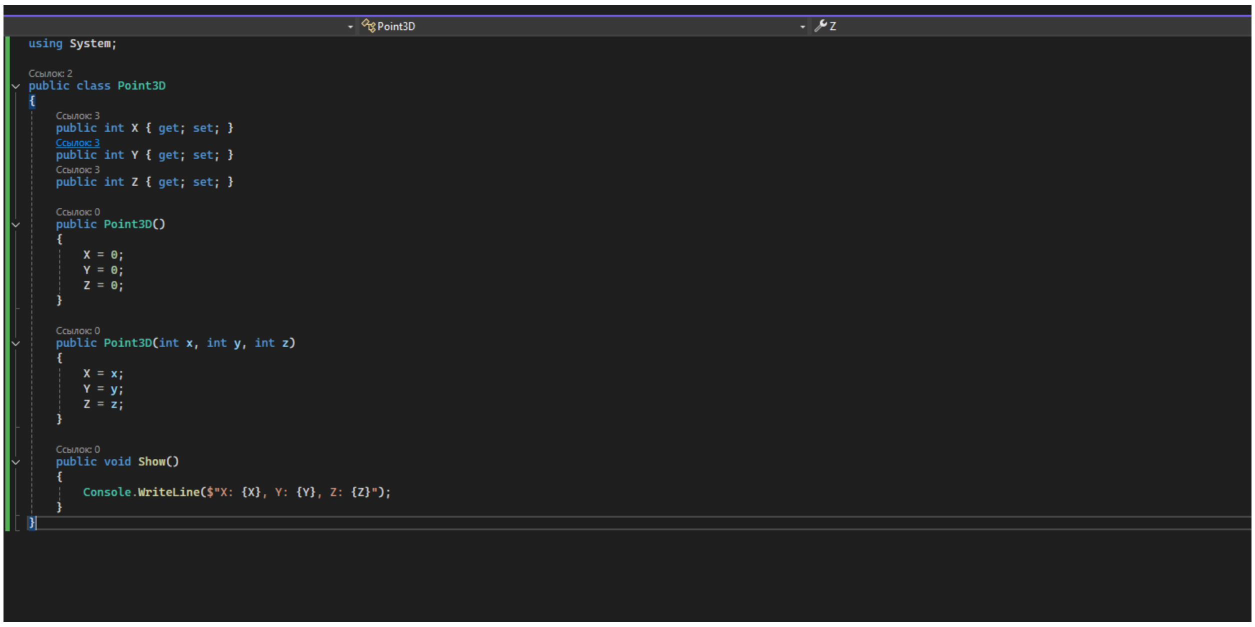
```
using System;

Ссылок: 2
public class Table
{
    Ссылок: 3
    public int Rows { get; set; }
    Ссылок: 3
    public int Cols { get; set; }

    Ссылок: 0
    public Table()
    {
        Rows = 0;
        Cols = 0;
    }

    Ссылок: 0
    public Table(int rows, int cols)
    {
        Rows = rows;
        Cols = cols;
    }

    Ссылок: 0
    public void Display()
    {
        Console.WriteLine($"Строки: {Rows}, Столбцы: {Cols}");
    }
}
```

2.6



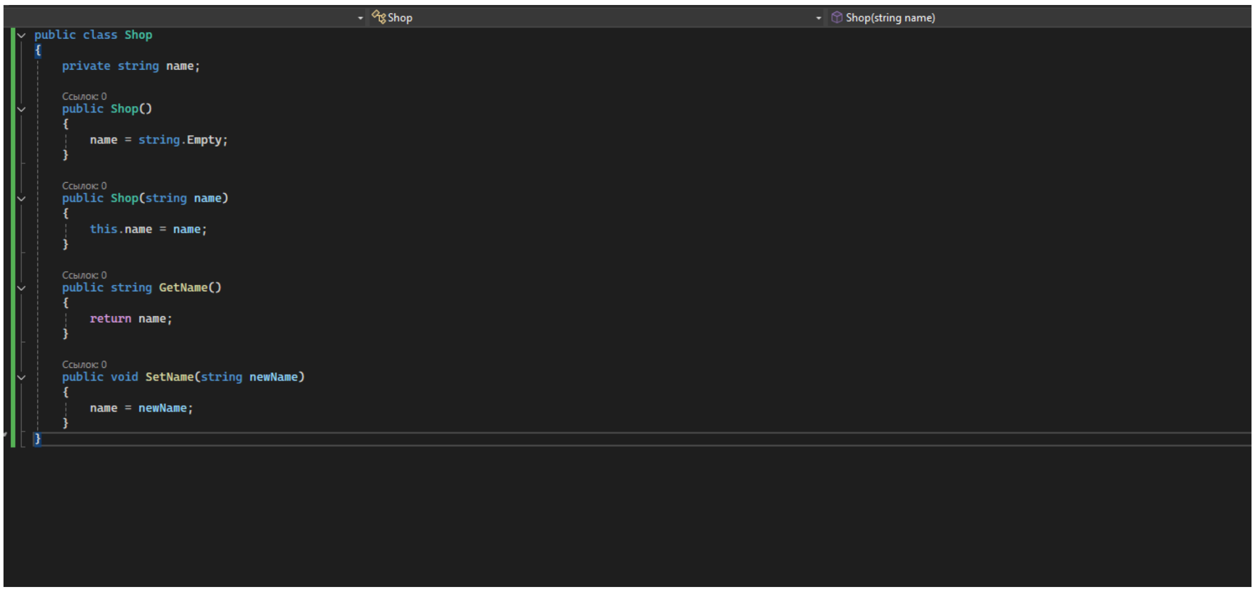
```
public class Manager
{
    private int age;
    private string name;

    Ссылок: 0
    public Manager()
    {
        age = 0;
        name = string.Empty;
    }

    Ссылок: 0
    public Manager(int age, string name)
    {
        this.age = age;
        this.name = name;
    }

    Ссылок: 0
    public int GetAge()
    {
        return age;
    }

    Ссылок: 0
    public string GetName()
    {
        return name;
    }
}
```

2.7



```
using System;

Ссылок: 2
public class Point3D
{
    Ссылок: 3
    public int X { get; set; }
    Ссылок: 3
    public int Y { get; set; }
    Ссылок: 3
    public int Z { get; set; }

    Ссылок: 0
    public Point3D()
    {
        X = 0;
        Y = 0;
        Z = 0;
    }

    Ссылок: 0
    public Point3D(int x, int y, int z)
    {
        X = x;
        Y = y;
        Z = z;
    }

    Ссылок: 0
    public void Show()
    {
        Console.WriteLine($"X: {X}, Y: {Y}, Z: {Z}");
    }
}
```

2.8



```
public class Shop
{
    private string name;

    Ссылок: 0
    public Shop()
    {
        name = string.Empty;
    }

    Ссылок: 0
    public Shop(string name)
    {
        this.name = name;
    }

    Ссылок: 0
    public string GetName()
    {
        return name;
    }

    Ссылок: 0
    public void SetName(string newName)
    {
        name = newName;
    }
}
```

3.1

```
using System;

public class Student
{
    public string Name { get; set; }
}

class Program
{
    static void Main()
    {
        Student student = new Student();
        student.Name = "Мария";
        Console.WriteLine($"Имя студента: {student.Name}");
    }
}
```

3.2

```
using System;
```

```
public class Child
{
    public int Age { get; set; }

    public Child()
    {
    }
}

class Program
{
    static void Main()
    {
        Child child = new Child { Age = 7 };
        Console.WriteLine($"Возраст ребенка: {child.Age}");
    }
}
```

3.3

```
using System;
```

```
public class Car
{
    private int year;

    public int Year
    {
        get { return year; }
        set
        {
            if (value > 0)
                year = value;
        }
    }

    public Car()
    {
    }
}

class Program
{
    static void Main()
    {
        Car car = new Car();
        car.Year = 2022;

        Console.WriteLine($"Год выпуска: {car.Year}");


        car.Year = -5;

        Console.WriteLine($"Год выпуска после попытки установить
отрицательное значение: {car.Year}");

    }

}
```

3.4

```
using System;
```

```
public class Car
{
    public string Name { get; set; }
    public string Color { get; set; }

    public Car()
    {
    }
}

class Program
{
    static void Main()
    {
        Car car = new Car { Name = "KIA SOUL", Color = "green" };
        Console.WriteLine($"Автомобиль: {car.Name}, Цвет: {car.Color}");
    }
}
```

3.5

```
using System;
```

```
}

public class Product
{
    protected string name;

    public string Name
    {
        get { return name; }
        private set { }
    }

    public Product()
    {
        name = "Рамиль";
    }
}

class Program
{
    static void Main()
    {
        Product product = new Product();
        Console.WriteLine($"Название продукта: {product.Name}");
    }
}
```

4.1

using System;

4.1

using System;

```
public class Person
{
    public string Name { get; set; }
    public int Age { get; set; }
}

public class Student : Person
{
    public string StudentId { get; set; }
    public string Major { get; set; }
}

class Program
{
    static void Main()
    {
        Student student = new Student();
        student.Name = "Анна";
        student.Age = 20;
        student.StudentId = "ST12345";
        student.Major = "Информатика";

        Console.WriteLine($"Студент: {student.Name}, Возраст: {student.Age}, ID: {student.StudentId}, Специальность: {student.Major}");
    }
}
```

4.2

```
using System;
```

```
public class Animal
{
    public string Name { get; set; }
    public int Age { get; set; }

    public void Eat()
    {
        Console.WriteLine($"{Name} ест");
    }

    public void Sleep()
    {
        Console.WriteLine($"{Name} спит");
    }
}

public class Cat : Animal
{
    public void Meow()
    {
        Console.WriteLine($"{Name} мяукает");
    }

    public void Purr()
    {
        Console.WriteLine($"{Name} мурлычет");
    }
}

public class Dog : Animal
{
    public void Bark()
    {
        Console.WriteLine($"{Name} лает");
    }

    public void WagTail()
    {
        Console.WriteLine($"{Name} виляет хвостом");
    }
}

class Program
{
    static void Main()
    {
        Cat cat = new Cat();
        cat.Name = "Барсик";
        cat.Age = 3;
        cat.Eat();

        cat.Sleep();

        cat.Meow();

        cat.Purr();


        Console.WriteLine();


        Dog dog = new Dog();

        dog.Name = "Рекс";

        dog.Age = 5;

        dog.Eat();

        dog.Sleep();

        dog.Bark();

        dog.WagTail();

    }

}
```

4.3

```
using System;
```

```
public class Entity
{
    public int Id { get; set; }
    public DateTime CreatedAt { get; set; }

    public Entity()
    {
        CreatedAt = DateTime.Now;
    }

    public void DisplayInfo()
    {
        Console.WriteLine($"ID: {Id}, Создано: {CreatedAt}");
    }
}

public class Product : Entity
{
    public string Name { get; set; }
    public decimal Price { get; set; }
    public int Quantity { get; set; }

    public void DisplayProductInfo()
    {
        Console.WriteLine($"Товар: {Name}, Цена: {Price:C}, Количество:
{Quantity}");
    }
}

class Program
{
    static void Main()
    {
        Product product = new Product();
        product.Id = 101;
        product.Name = "Ноутбук";
        product.Price = 75000;
        product.Quantity = 10;

        product.DisplayInfo();
        product.DisplayProductInfo();
    }
}
```

4.4

```
using System;
```

```
public class Dishes
{
    public string Material { get; set; }
    public string Color { get; set; }
    public double Weight { get; set; }

    public void Wash()
    {
        Console.WriteLine("Посуда моется");
    }

    public void Dry()
    {
        Console.WriteLine("Посуда сушится");
    }
}

public class Cup : Dishes
{
    public double Volume { get; set; }
    public bool HasHandle { get; set; }

    public void Fill()
    {
        Console.WriteLine("Чашка наполняется");
    }

    public void Drink()
    {
        Console.WriteLine("Пьем из чашки");
    }
}

class Program
{
    static void Main()
    {
        Cup cup = new Cup();
        cup.Material = "Керамика";
        cup.Color = "Белый";
        cup.Weight = 0.3;
        cup.Volume = 250;
        cup.HasHandle = true;

        cup.Wash();
        cup.Dry();
        cup.Fill();
        cup.Drink();
        Console.WriteLine($"Чашка: материал - {cup.Material}, цвет -
{cup.Color}, объем - {cup.Volume}мл");
    }
}
```

4.5

```
using System;
```

```
public class Entity
{
    public int Id { get; set; }
    public string Name { get; set; }

    public Entity()
    {
        Id = 0;
        Name = "Не указано";
    }

    public void DisplayEntityInfo()
    {
        Console.WriteLine($"Entity ID: {Id}, Имя: {Name}");
    }
}

public class Staff : Entity
{
    public string Position { get; set; }
    public decimal Salary { get; set; }

    public Staff()
    {
        Position = "Сотрудник";
        Salary = 0;
    }

    public void Work()
    {
        Console.WriteLine($"{Name} работает на должности {Position}");
    }
}

public class Manager : Staff
{
    public int TeamSize { get; set; }
    public string Department { get; set; }

    public Manager()
    {
        TeamSize = 0;
        Department = "Не указан";
    }

    public void ManageTeam()
    {
        Console.WriteLine($"{Name} управляет командой из {TeamSize} человек
в отделе {Department}");
    }
    public void ConductMeeting()
    {
        Console.WriteLine($"{Name} проводит собрание");
    }
}

class Program
{
    static void Main()
    {
        Manager manager = new Manager();
        manager.Id = 1001;
        manager.Name = "Алексей Петров";
        manager.Position = "Руководитель отдела";
        manager.Salary = 150000;
        manager.TeamSize = 8;
        manager.Department = "Разработка";

        manager.DisplayEntityInfo();
        manager.Work();
        manager.ManageTeam();
        manager.ConductMeeting();
    }
}
```

4.6

4.6

```
using System;
```

```
public class Animal
{
    protected int age;

    public Animal()
    {
        age = 0;
    }

    public void SetAge(int newAge)
    {
        if (newAge > 0)
            age = newAge;
    }

    public void ShowAge()
    {
        Console.WriteLine($"Возраст животного: {age}");
    }
}

public class Predator : Animal
{
    public void Hunt()
    {
        Console.WriteLine($"Хищник в возрасте {age} лет охотится");
    }

    public void IncreaseAge()
    {
        age++;
        Console.WriteLine($"Возраст хищника увеличен. Теперь: {age}");
    }
}

class Program
{
    static void Main()
    {
        Predator predator = new Predator();
        predator.SetAge(5);
        predator.ShowAge();
        predator.Hunt();
        predator.IncreaseAge();
        predator.Hunt();
    }
}
```

4.7

```
using System;
```

```
public class Transport
{
    protected string name;

    public Transport()
    {
        name = "Неизвестный транспорт";
    }

    public void SetName(string newName)
    {
        name = newName;
    }

    public void ShowName()
    {
        Console.WriteLine($"Название транспорта: {name}");
    }
}

public class SpaceShuttle : Transport
{
    public void Launch()
    {
        Console.WriteLine($"Космический корабль '{name}' запущен в космос");
    }

    public void SetShuttleName(string shuttleName)
    {
        name = shuttleName;
        Console.WriteLine($"Кораблю присвоено имя: {name}");
    }
}

class Program
{
    static void Main()
    {
        SpaceShuttle shuttle = new SpaceShuttle();
        shuttle.SetShuttleName("Атлантис");
        shuttle.ShowName();
        shuttle.Launch();
    }
}
```

5.1

```
using System;
```

```
public class Strategy
{
    public virtual void Display()
    {
        Console.WriteLine("Strategy");
    }
}

class Program
{
    static void Main()
    {
        Strategy strategy = new Strategy();
        strategy.Display();
    }
}
```

5.2

```
using System;
```

```
public class Weather
{
    public virtual void Show()
    {
        Console.WriteLine("My Weather");
    }
}

class Program
{
    static void Main()
    {
        Weather weather = new Weather();
        weather.Show();
    }
}
```

5.3

```
using System;
```

```
public class Strategy

{

    public virtual void Display()

    {

        Console.WriteLine("Strategy");

    }

}


public class ConservativeStrategy : Strategy

{

    public override void Display()

    {

        Console.WriteLine("ConservativeStrategy");

    }

}


class Program

{

    static void Main()

    {

        Strategy strategy = new Strategy();

        strategy.Display();


        ConservativeStrategy conservative = new ConservativeStrategy();
        conservative.Display();


        Strategy polyStrategy = new ConservativeStrategy();
        polyStrategy.Display();
    }
}
```

5.4

```
using System;
```

```
public class Animal
{
    private string type;

    public Animal()
    {
        type = "My Type";
    }

    public virtual void Print()
    {
        Console.WriteLine($"Тип животного: {type}");
    }
}

public class Cat : Animal
{
    private int age;

    public Cat()
    {
        age = 5;
    }
    public override void Print()
    {
        Console.WriteLine($"Возраст кота: {age}");
    }
}

class Program
{
    static void Main()
    {
        Animal animal = new Animal();
        animal.Print();

        Cat cat = new Cat();
        cat.Print();

        Animal animalCat = new Cat();
        animalCat.Print();
    }
}
```

5.5

```
using System;
```

```
public abstract class Entity
{
    public abstract void Display();
}

class Program
{
    static void Main()
    {
        Console.WriteLine("Абстрактный класс Entity содержит абстрактный метод Display");
    }
}
```

5.6

```
using System;
```

```
public abstract class Entity
{
    public abstract void Display();
}

public class Product : Entity
{
    public override void Display()
    {
        Console.WriteLine("My Product");
    }
}

class Program
{
    static void Main()
    {
        Product product = new Product();
        product.Display();
    }
}
```

5.7

```
using System;
```

```
public interface IPrintable
{
    void Display();
}

public class ConsolePrinting : IPrintable
{
    public void Display()
    {
        Console.WriteLine("My Console");
    }
}

class Program
{
    static void Main()
    {
        ConsolePrinting consolePrinting = new ConsolePrinting();
        consolePrinting.Display();
    }
}
```