



**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И  
НАУКИ  
ГОРОДА МОСКВЫ  
Государственное бюджетное  
профессиональное  
образовательное учреждение города  
Москвы  
«Колледж малого бизнеса № 4»  
(ГБПОУ КМБ № 4)**

**Отчёт по Домашнему заданию (Основы Алгоритмизации  
и программирования).**

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент(ка): Геронок Артём Игоревич

Группа: ИПО-22.24

Руководитель: Рыбаков Александр Сергеевич

Отчётная работа защищена с оценкой «\_\_\_» \_\_\_\_\_

Москва, 2026 г.



**ЗАДАНИЕ:****Основные понятия объектно-ориентированного программирования.****1. Инкапсуляция**

Практика:

1 Используя , напишите программный код следующих заданий.

1.1. Создайте класс Student. Определите в этом классе строковое поле name (имя).

1.2. Создайте класс Car. Определите в этом классе целочисленное поле year (год выпуска).

1.3. Реализуйте класс Point. Определите в этом классе целочисленное поле x.

1.4. Создайте класс Person с целочисленным полем age (возраст). Реализуйте в этом классе метод Print для печати поля age в консольном окне.

1.5. Разработайте класс Table с целочисленными полями rows (строки), cols (столбцы). Реализуйте в этом классе метод Display для отображения полей rows, cols в консольном окне.

1.6. Объявите класс Manager с полями age (целочисленный тип, возраст) и name (строковый тип, имя). Создайте метод GetAge для получения значения поля age и метод GetName для получения значения поля name.

1.7. Создайте класс Point3D с целочисленными полями x, y, z. Реализуйте метод Show для отобра-

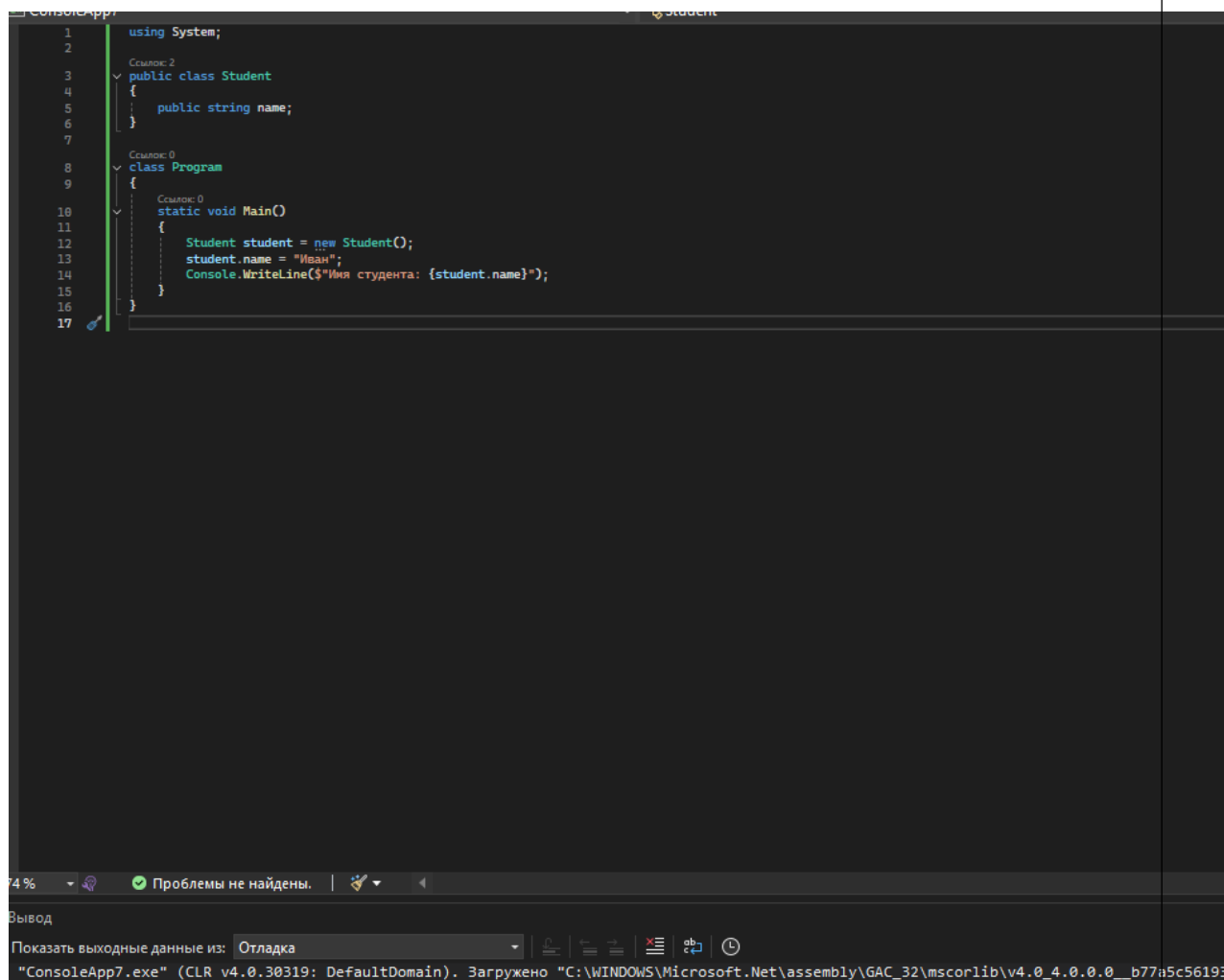
жения полей x, y, z в консольном окне.

1.8. Объявите класс Shop с строковым полем name (имя). В этом классе реализуйте метод GetName

для

получения значения поля name и метод SetName для установления нового значения полю name.

1.1



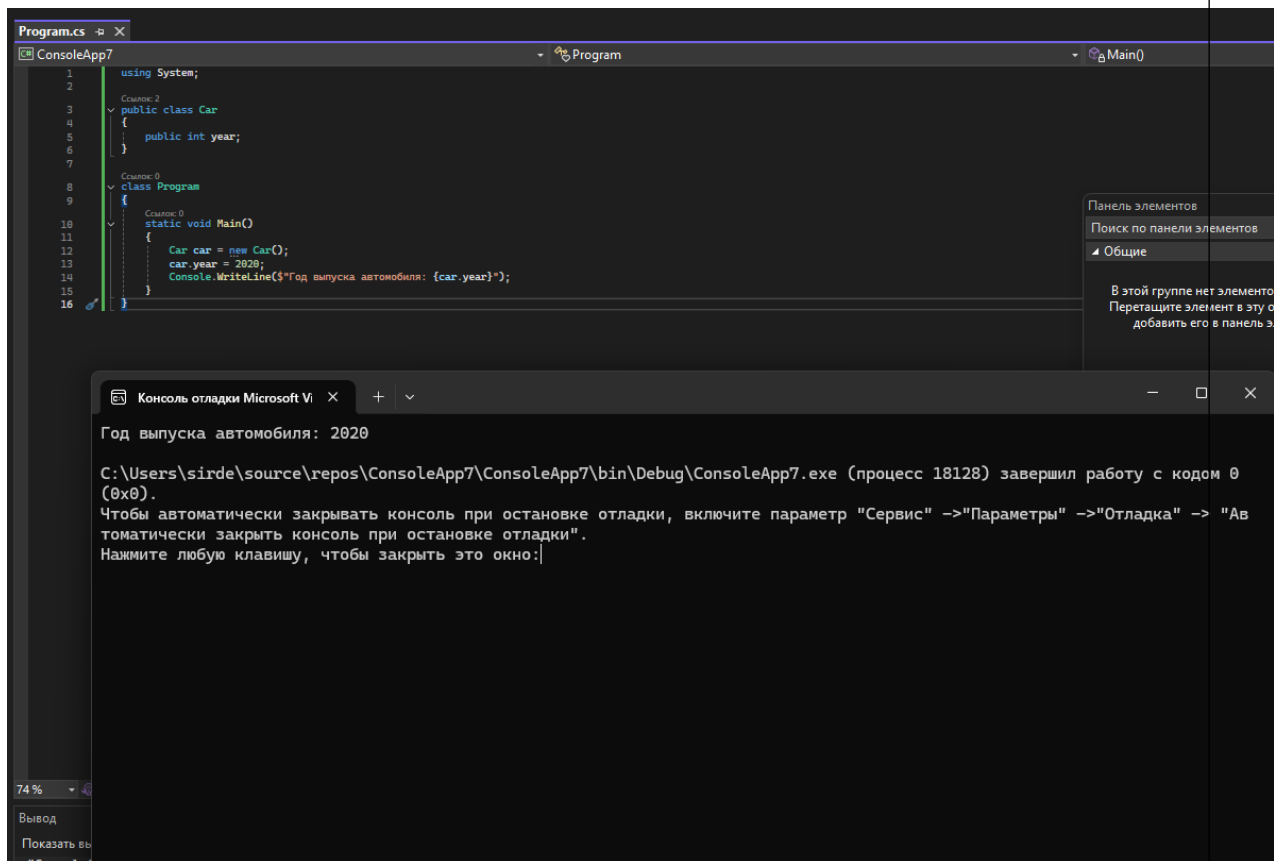
```
1 using System;
2
3 Ссылка: 2
4 public class Student
5 {
6     public string name;
7 }
8
9 Ссылка: 0
10 class Program
11 {
12     Ссылка: 0
13     static void Main()
14     {
15         Student student = new Student();
16         student.name = "Иван";
17         Console.WriteLine($"Имя студента: {student.name}");
18     }
19 }
```

Вывод

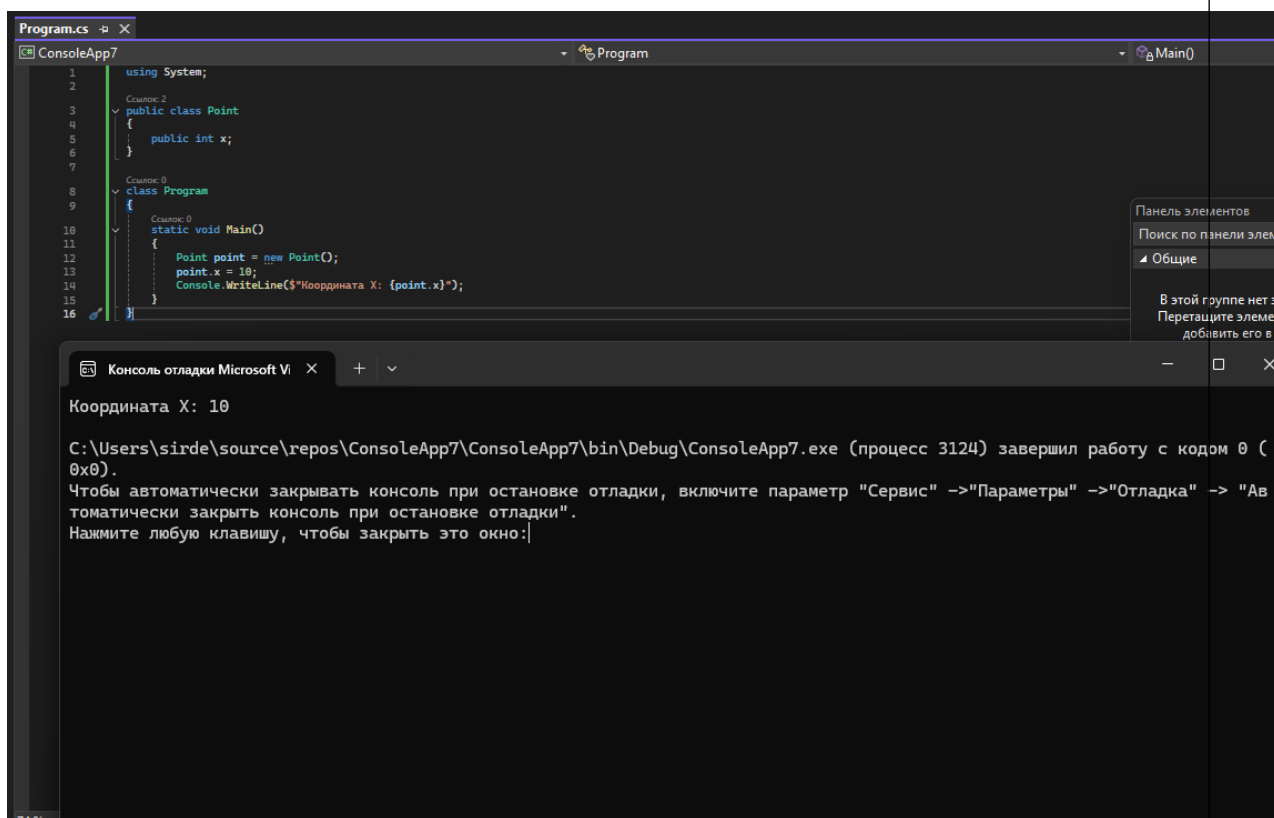
Показать выходные данные из: Отладка

"ConsoleApp7.exe" (CLR v4.0.30319: DefaultDomain). Загружено "C:\WINDOWS\Microsoft.Net\assembly\GAC\_32\mscorlib\v4.0.0.0\_b77a5c561934e089\_x-ww\_3935f2c0\mscorlib.dll"

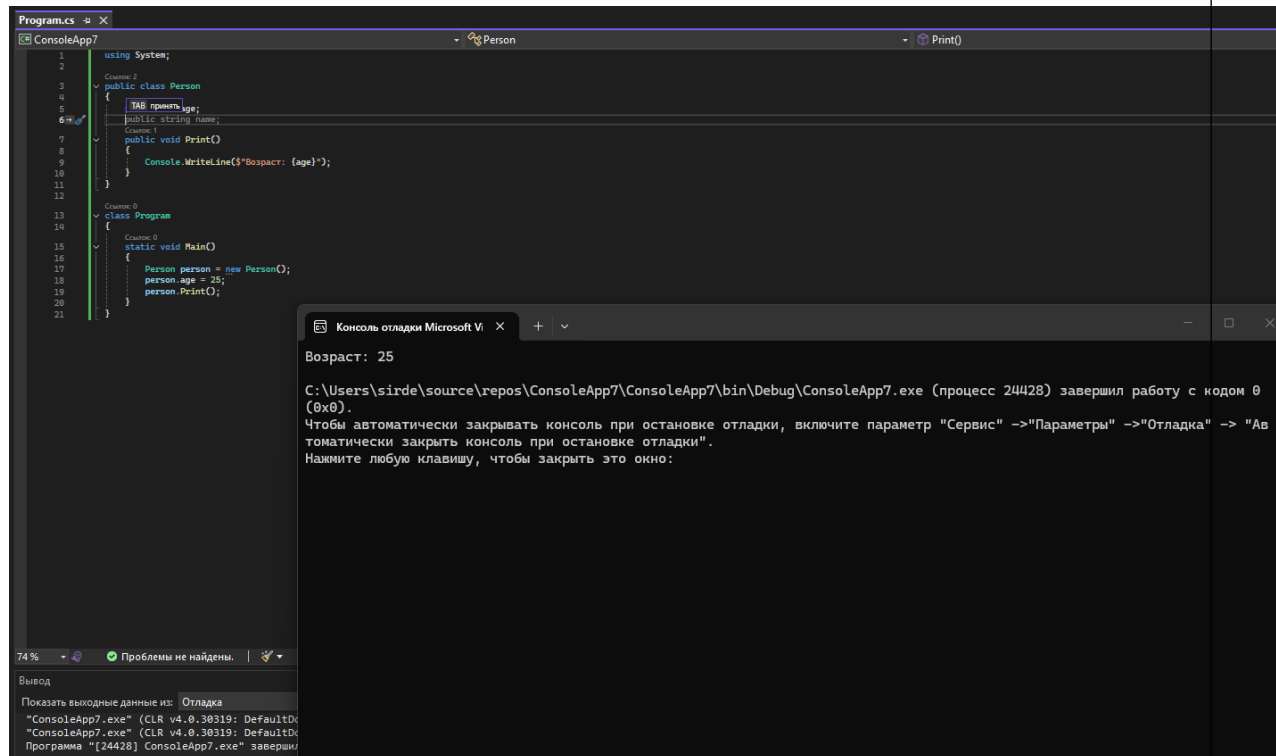
1.2



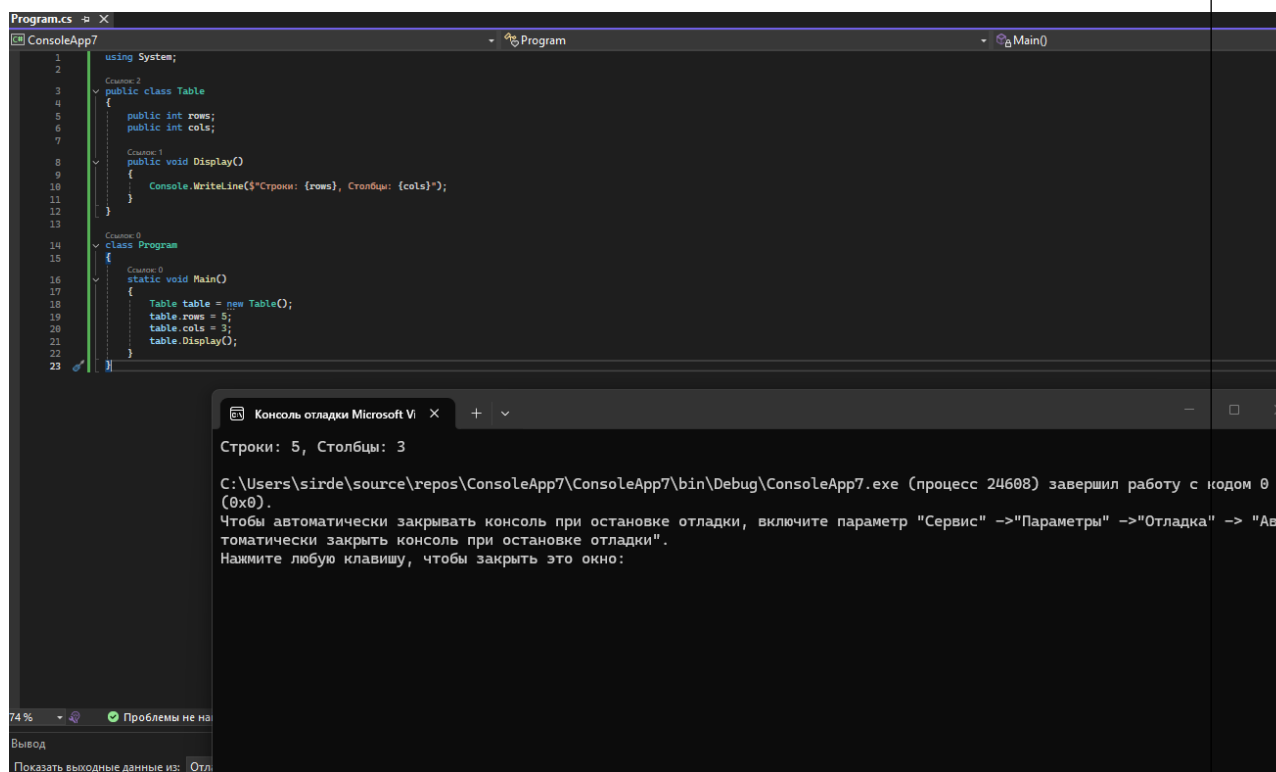
### 1.3



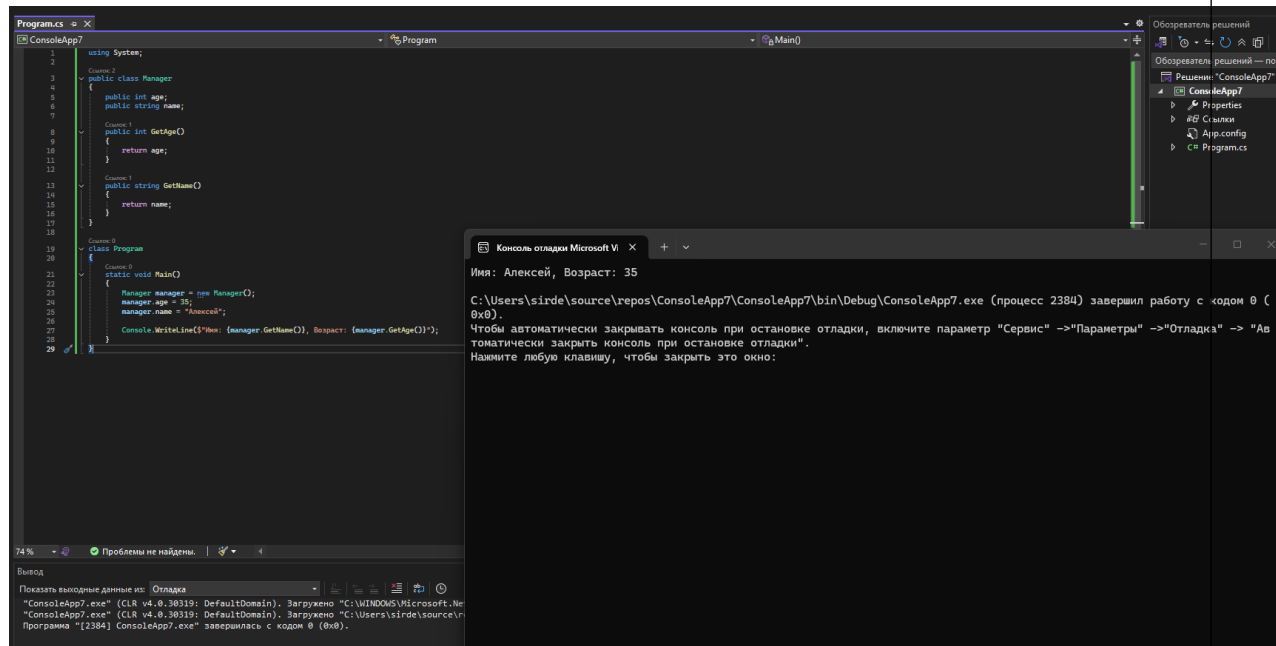
### 1.4



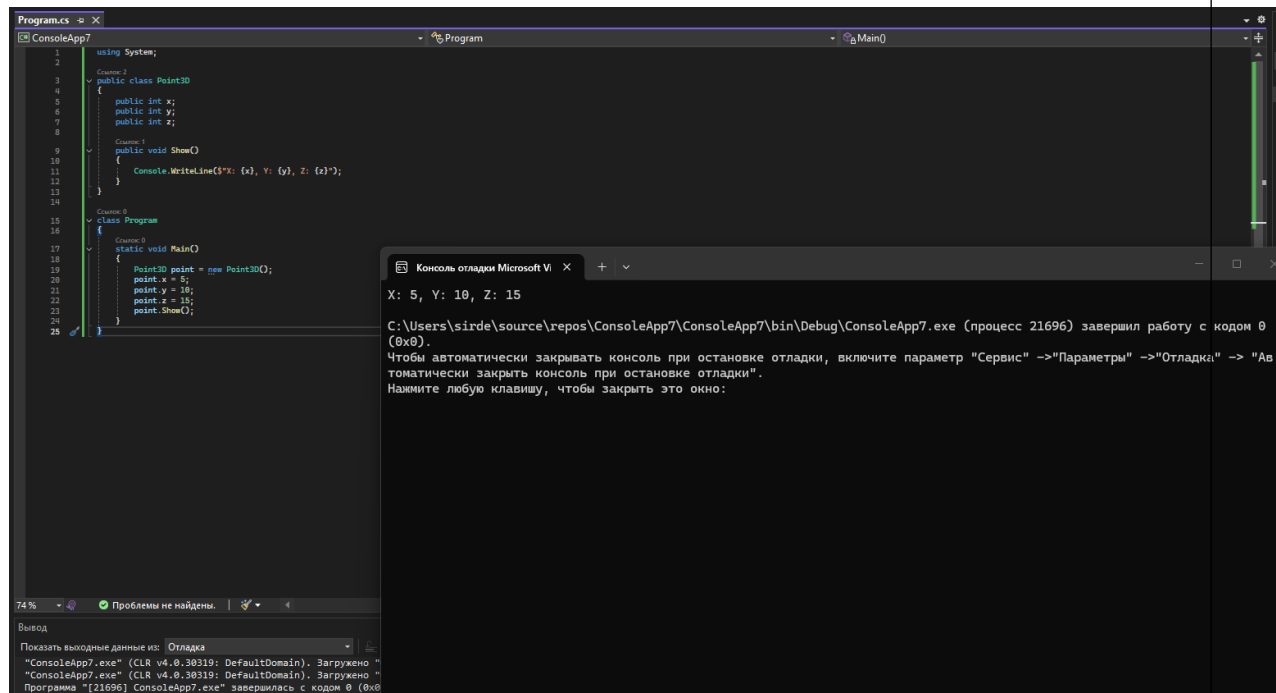
1.5



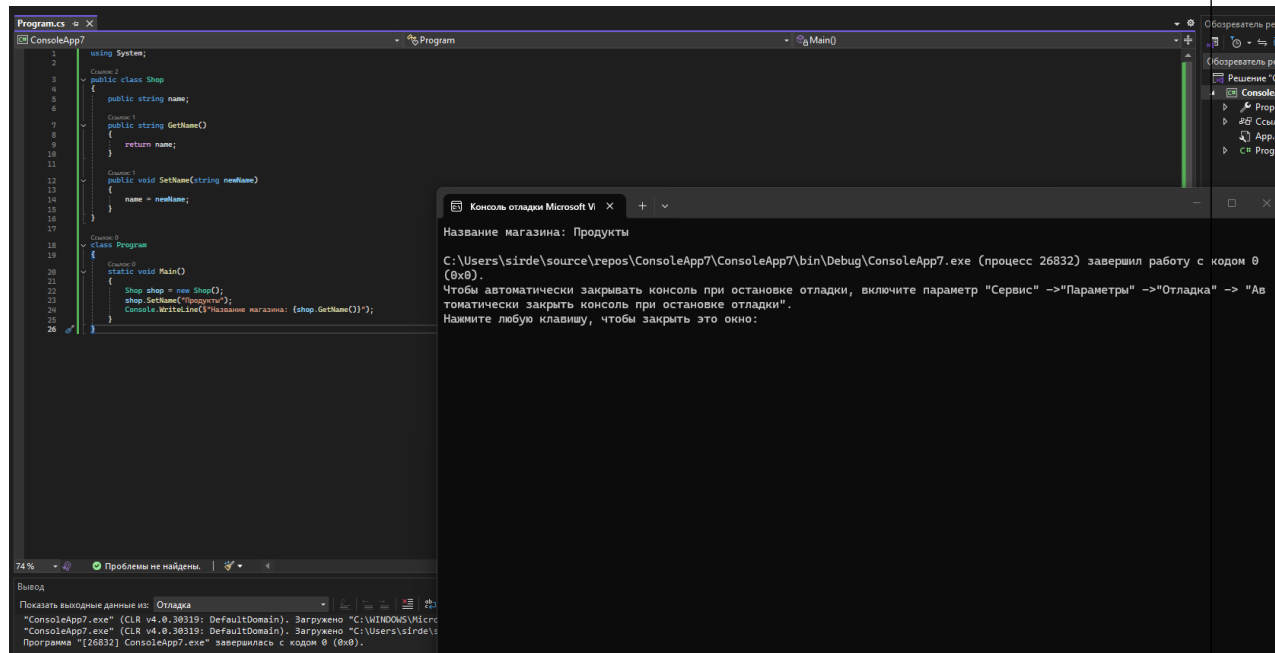
1.6



## 1.7



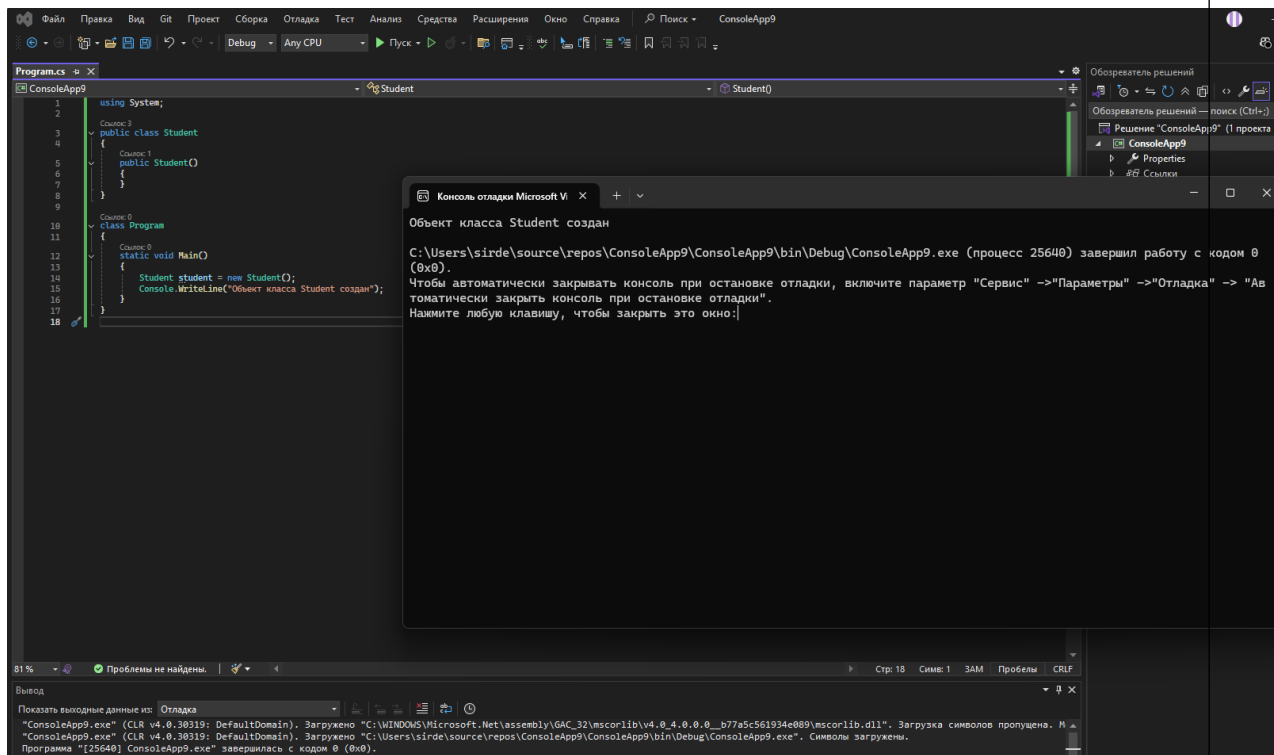
## 1.8



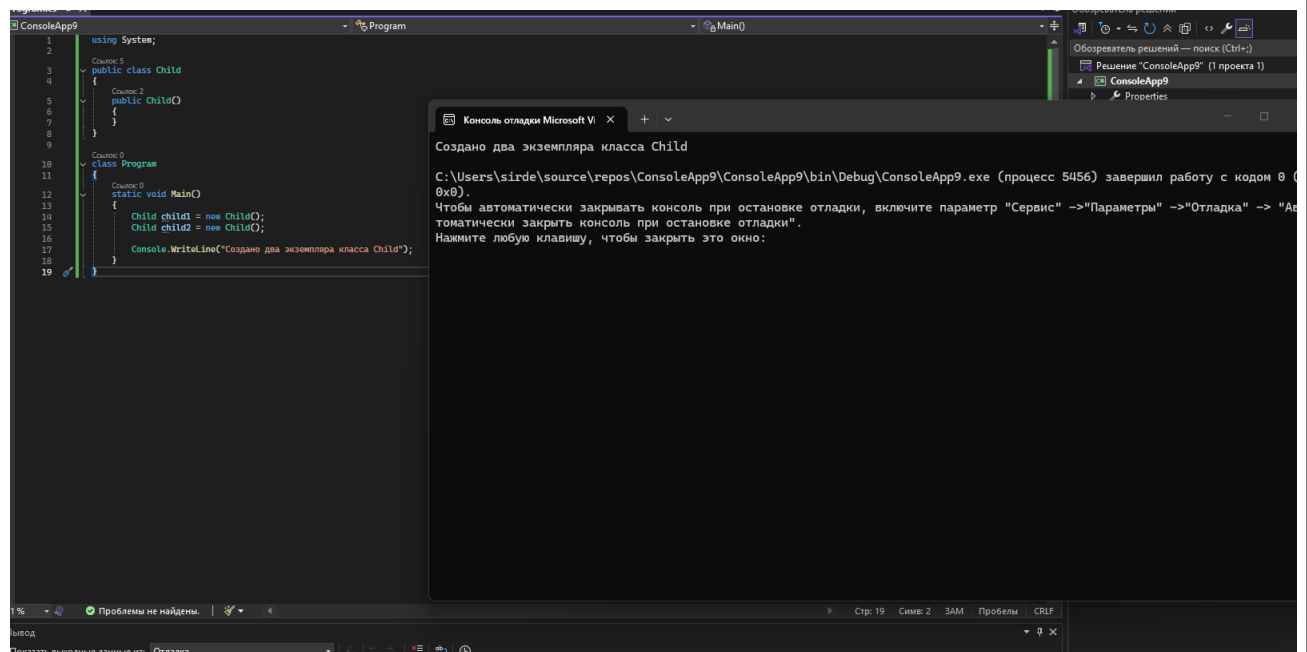
## 2. Конструкторы класса

# Конструкторы, инициализаторы и деконструкторы

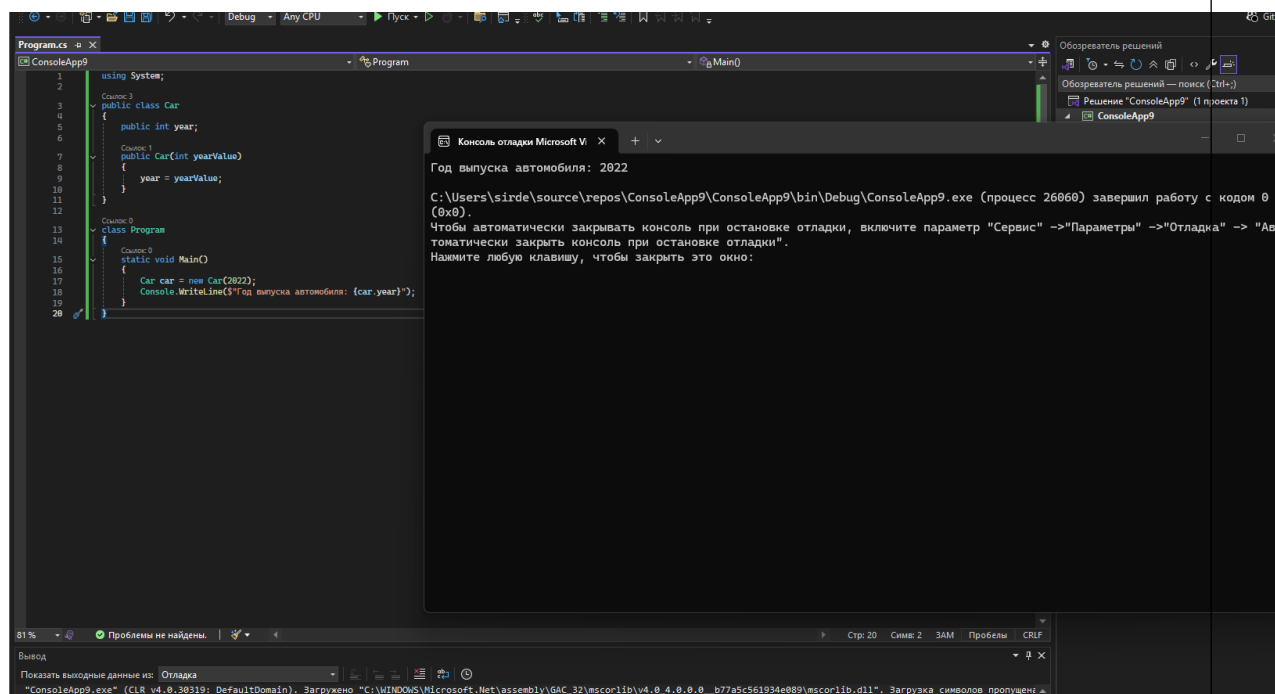
### 2.1.



### 2.2.



## 2.3.



**2.4. Создайте класс Person с целочисленным полем age (возраст). Реализуйте в этом классе метод Print для печати поля age в консольном окне.**

```

1 using System;
2
3 public class Car
4 {
5     public string name;
6     public string color;
7
8     public Car(string nameValue, string colorValue)
9     {
10         name = nameValue;
11         color = colorValue;
12     }
13
14 }
15
16 class Program
17 {
18     static void Main()
19     {
20         Car lada = new Car("LADA VESTA", "black");
21         Car bmw = new Car("BMW X5", "white");
22
23         Console.WriteLine($"Автомобиль: {lada.name}, Цвет: {lada.color}");
24         Console.WriteLine($"Автомобиль: {bmw.name}, Цвет: {bmw.color}");
25     }
26 }

```

Консоль отладки Microsoft Visual Studio

Автомобиль: LADA VESTA, Цвет: black  
Автомобиль: BMW X5, Цвет: white

C:\Users\sirde\source\repos\ConsoleApp9\ConsoleApp9\bin\Debug\ConsoleApp9.exe (процесс 22856) завершил работу с кодом (0x0).

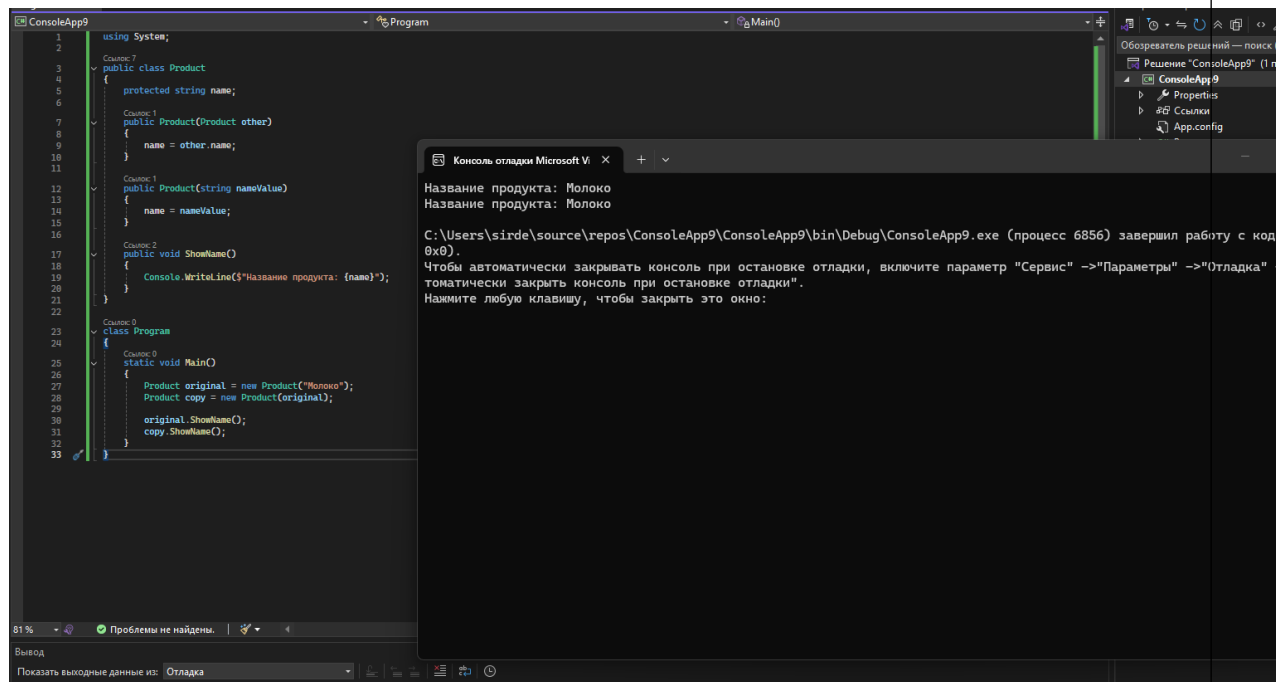
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрывать консоль при остановке отладки".  
Нажмите любую клавишу, чтобы закрыть это окно:

Вывод

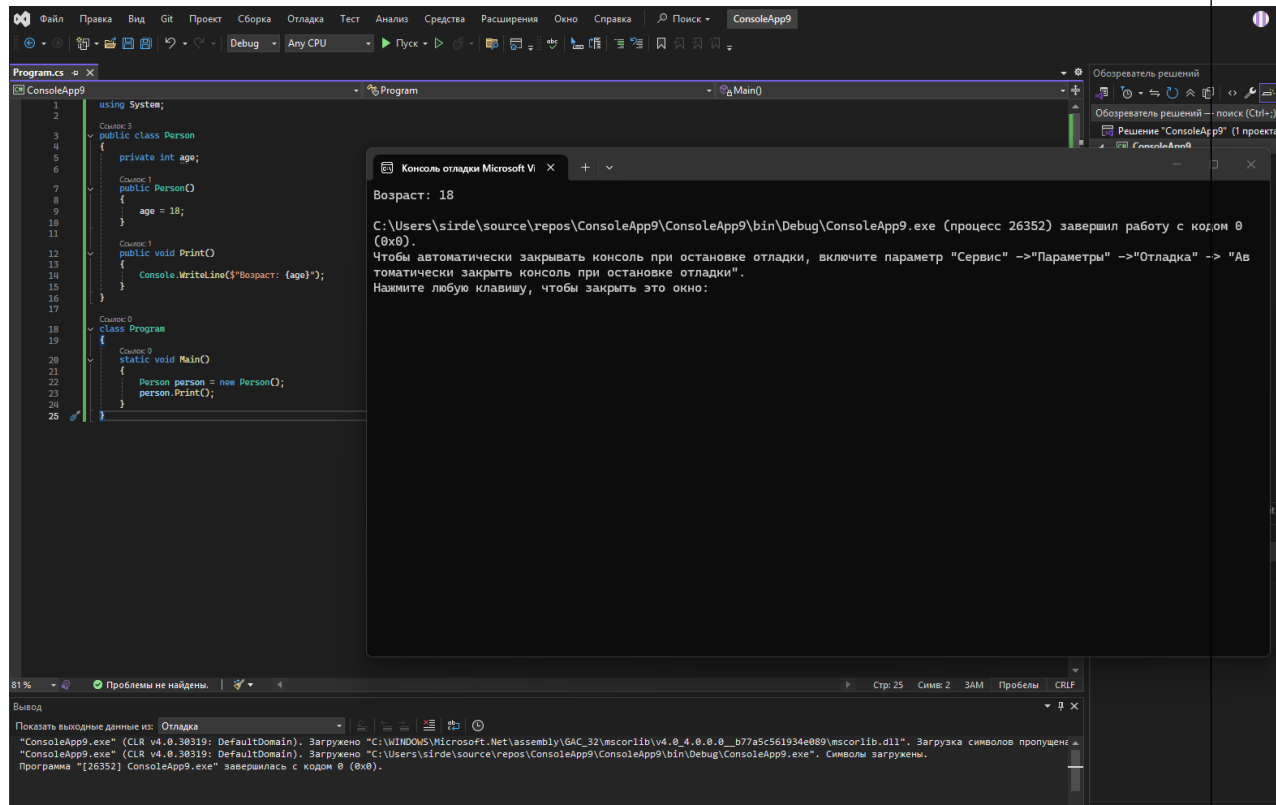
Показать выходные данные из: Отладка

"ConsoleApp9.exe" (CLR v4.0.30319: DefaultDomain). Загружено "C:\WINDOWS\Microsoft.NET\assembly\GAC\_32\mscorlib\v4.0.0.0\_b77a5c561934e089\mscorlib.dll". Загрузка символов пропущена.  
"ConsoleApp9.exe" (CLR v4.0.30319: DefaultDomain). Загружено "C:\Users\sirde\source\repos\ConsoleApp9\ConsoleApp9\bin\Debug\ConsoleApp9.exe". Символы загружены.

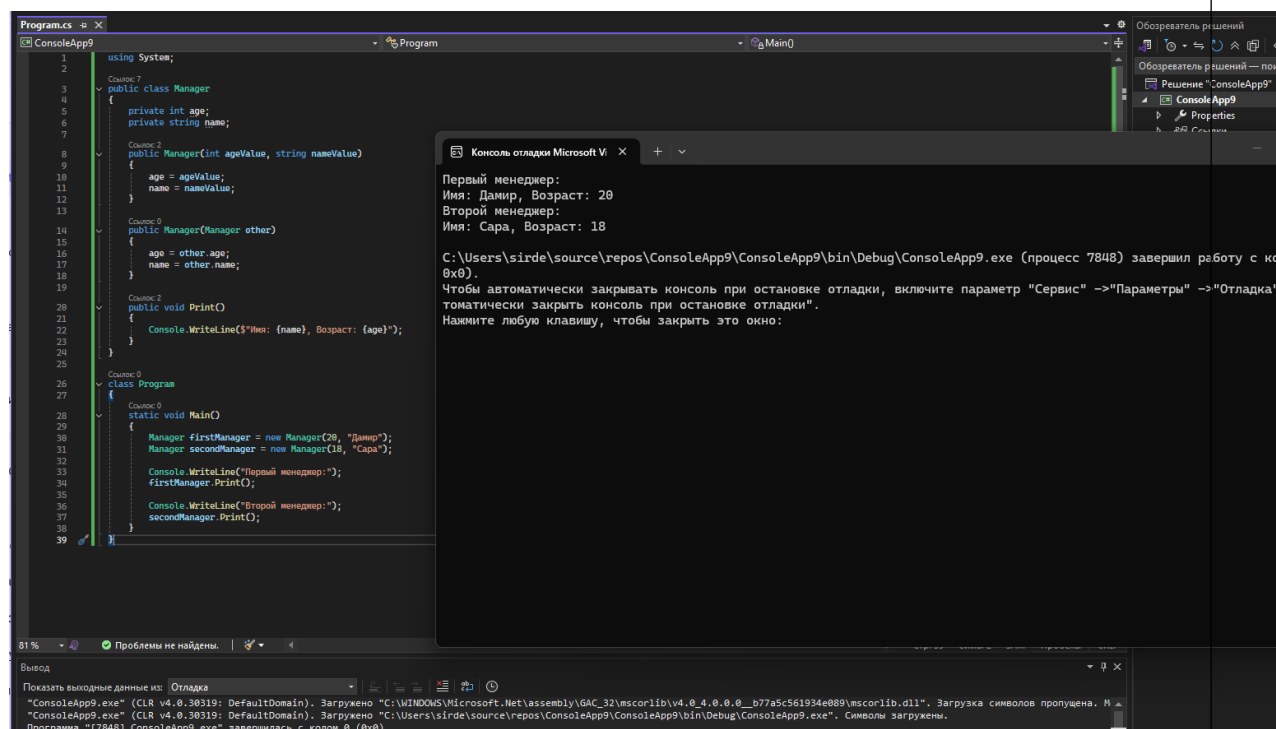
**2.5. Разработайте класс Table с целочисленными полями rows (строки), cols (столбцы). Реализуйте в этом классе метод Display для отображения полей rows, cols в консольном окне.**



**2.6. Объявите класс Manager с полями age (целочисленный тип, возраст) и name (строковый тип, имя). Создайте метод GetAge для получения значения поля age и метод GetName для получения значения поля name.**



2.7





### 3 Свойства классов

Практика:

3 Используя, напишите программный код следующих заданий.

3.1. Создайте класс `Student`. Определите в этом классе строковое свойство `Name` (имя).

3.2. Создайте класс `Child` с открытыми стандартным конструктором и целочисленным свойством `Age` (возраст).

Создайте

экземпляра класса `Child` с инициализацией свойства `Age`.

3.3. Создайте класс `Car`. Определите в этом классе открытое целочисленное

стандартное

свойство `Year`, закрытое целочисленное поле `year` (год выпуска) и общедоступный стандартный конструктор. В свойстве `Year` доступ к полю `year` осуществляется без ограничений, а изменение значения поля `year` допустимо на положительные числа.

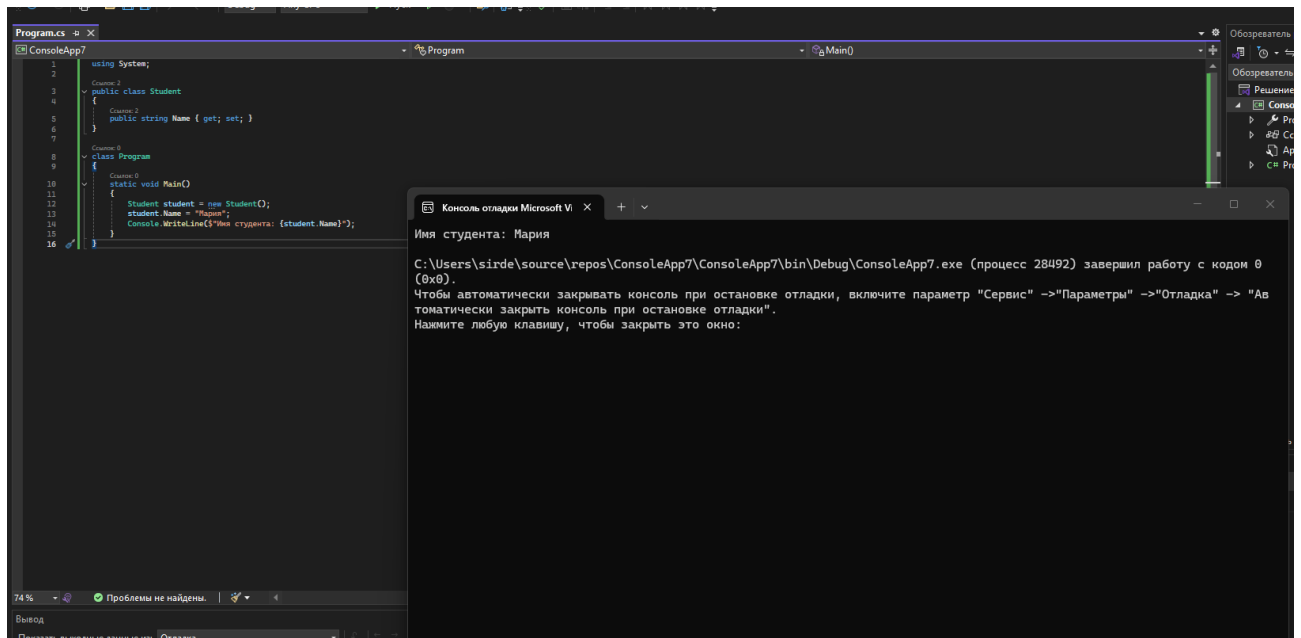
3.4. Определите класс `Car` с открытыми строковыми свойствами `Name` (имя) и `Color` (цвет) и с открытым стандартным конструктором. Используйте объектную инициализацию

создания экземпляра класса `Car`. В создаваемом объекте свойству `Name` присвойте значение `"KIA SOUL"`, а

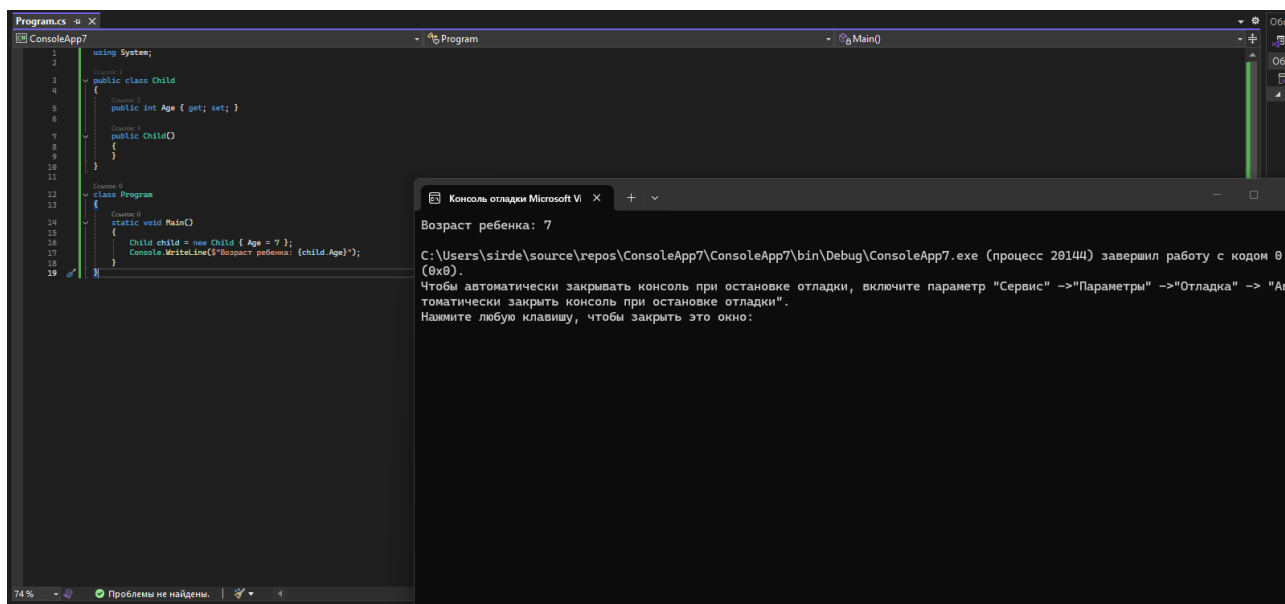
свойству Color – значение «green».

3.5. Создайте класс Product с защищенным строковым полем name. В этом классе реализуйте открытое свойство Name с ограниченным доступом. В свойстве Name невозможно изменять значение поля name, но можно получить доступ на чтение к полю name. Кроме того, создайте открытый стандартный конструктор с присвоением полю name значения «Рамиль».

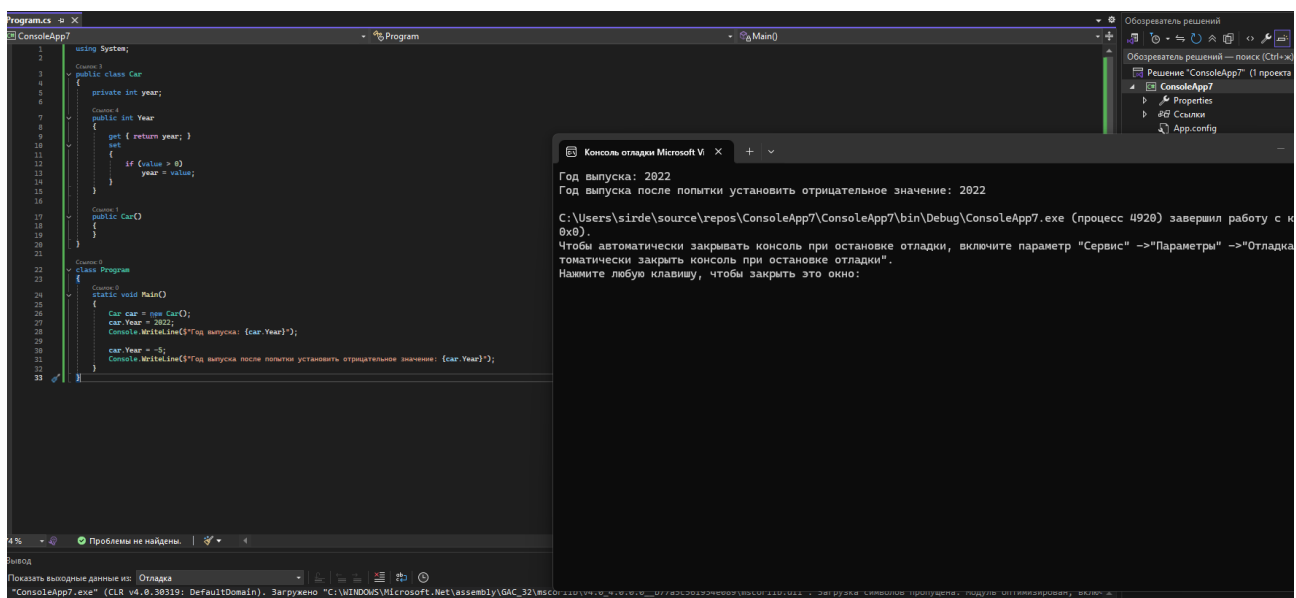
### 3.1



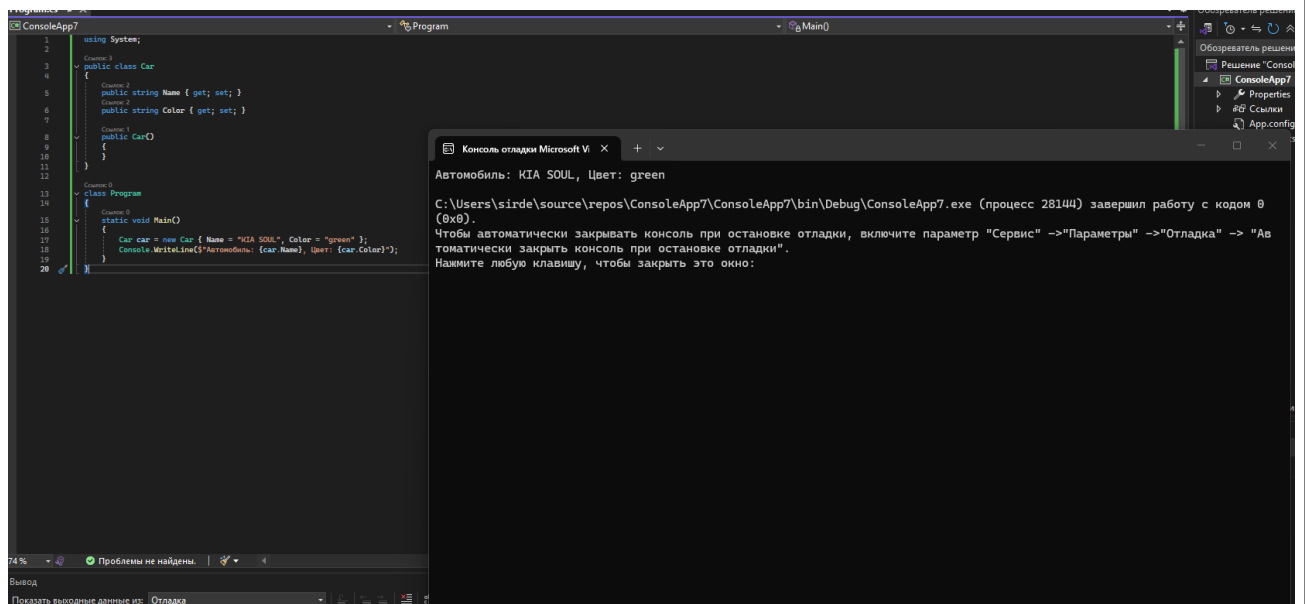
### 3.2



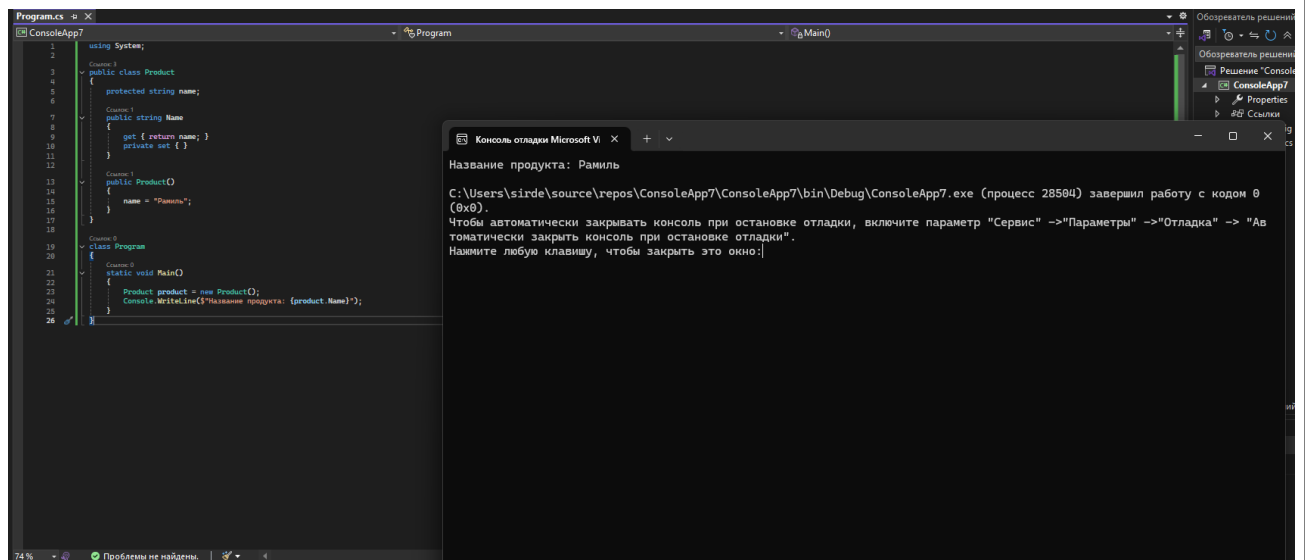
### 3.3



### 3.4



### 3.5



## 4 Наследование

Практика:

4. Используя, напишите программный код следующих заданий.

4.1. Создайте два класса Person и Student. Класс Student является дочерним, а класс Person – базовым. Установите корректное отношение наследования между классами Person и Student.

4.2. Создайте три класса Animal, Cat, Dog. Класс Animal является родительским для классов Cat, Dog. Установите правильные отношения наследования между этими тремя классами.

4.3. Создайте два класса Entity, Product. Entity является базовым классом, а Product – производным. Установите корректное

отношение

ния между двумя классами Entity, Product.

4.4. Пусть класс Dishes является родительским, а Cup – дочерним классом. Установить корректное отношение наследования между двумя классами Dishes, Cup.

4.5. Создайте три класса Entity, Staff, Manager. Класс Manager наследуется от класса Staff, а класс Staff от класса Entity. Установите правильные отношения наследования между этими классами.

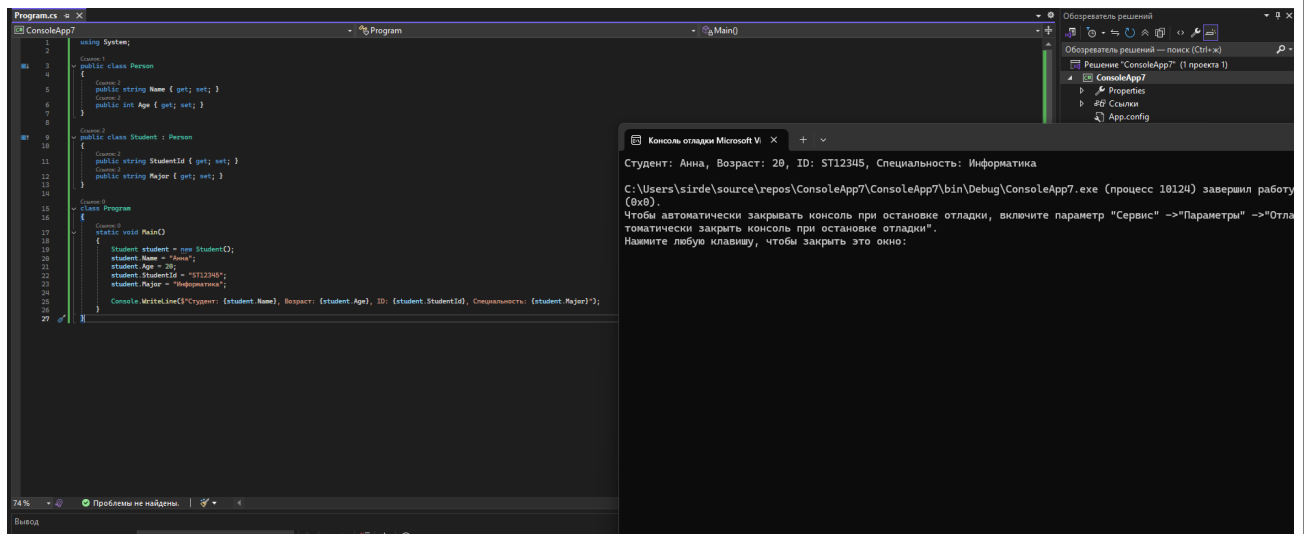
4.6. Реализуйте два класса Animal, Predator, где Animal – родительский класс, а Predator – дочерний. Создайте целочисленное поле age. Доступ к полю age возможен в контекстах классов Animal, Predator.

4.7. Пусть класс Transport является базовым, а SpaceShuttle – дочерний

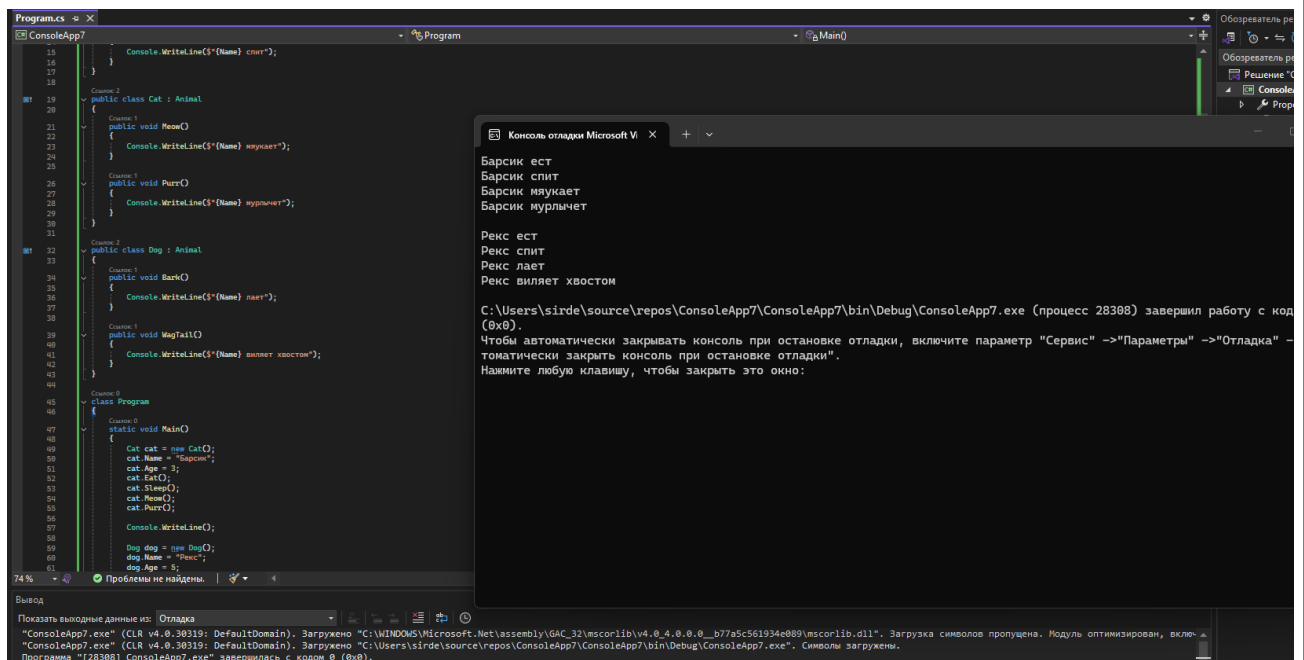
класс.

строковое поле name в рамках классов Transport, SpaceShuttle.

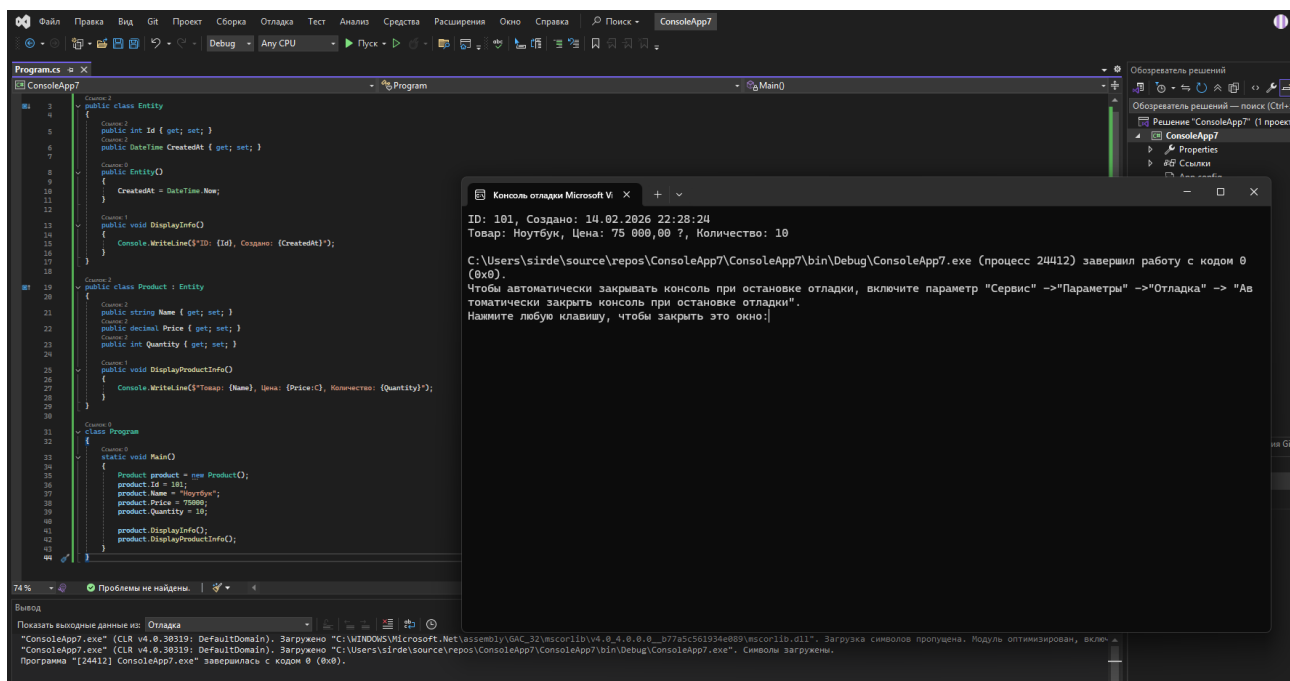
## 4.1



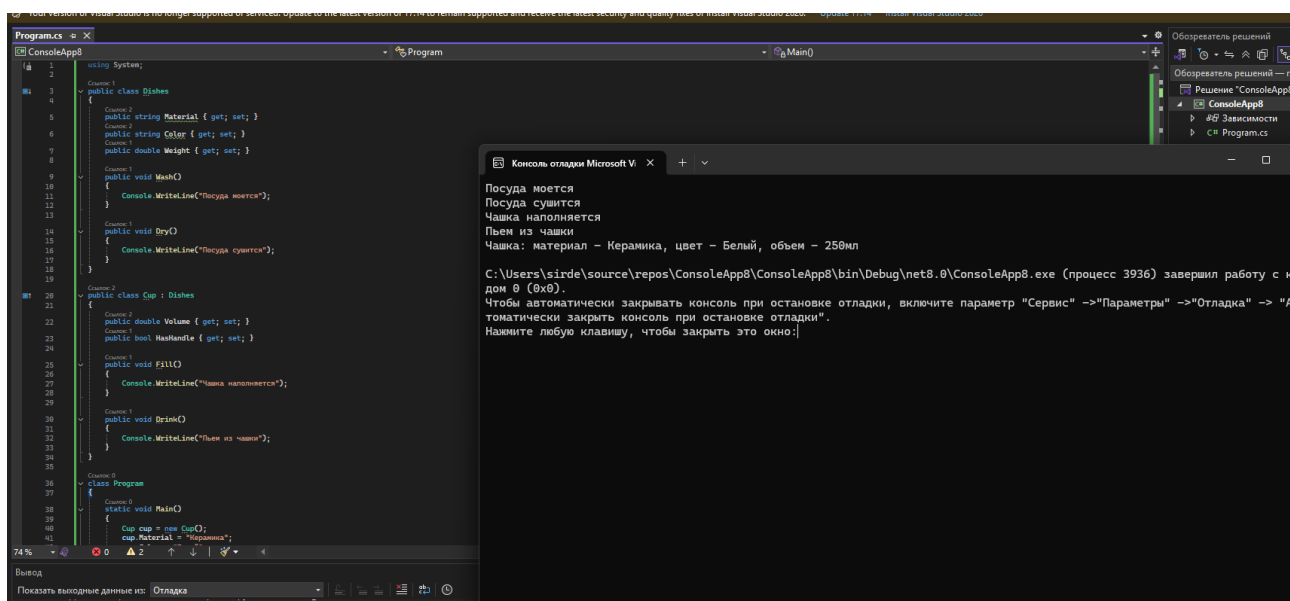
## 4.2



## 4.3

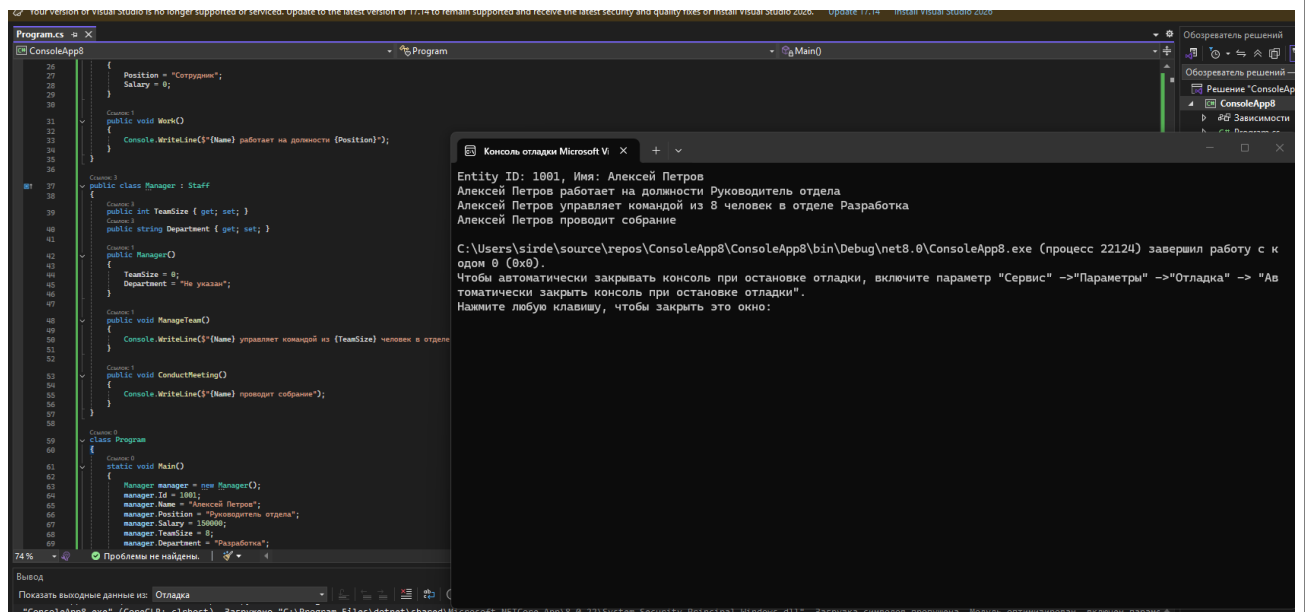


## 4.4

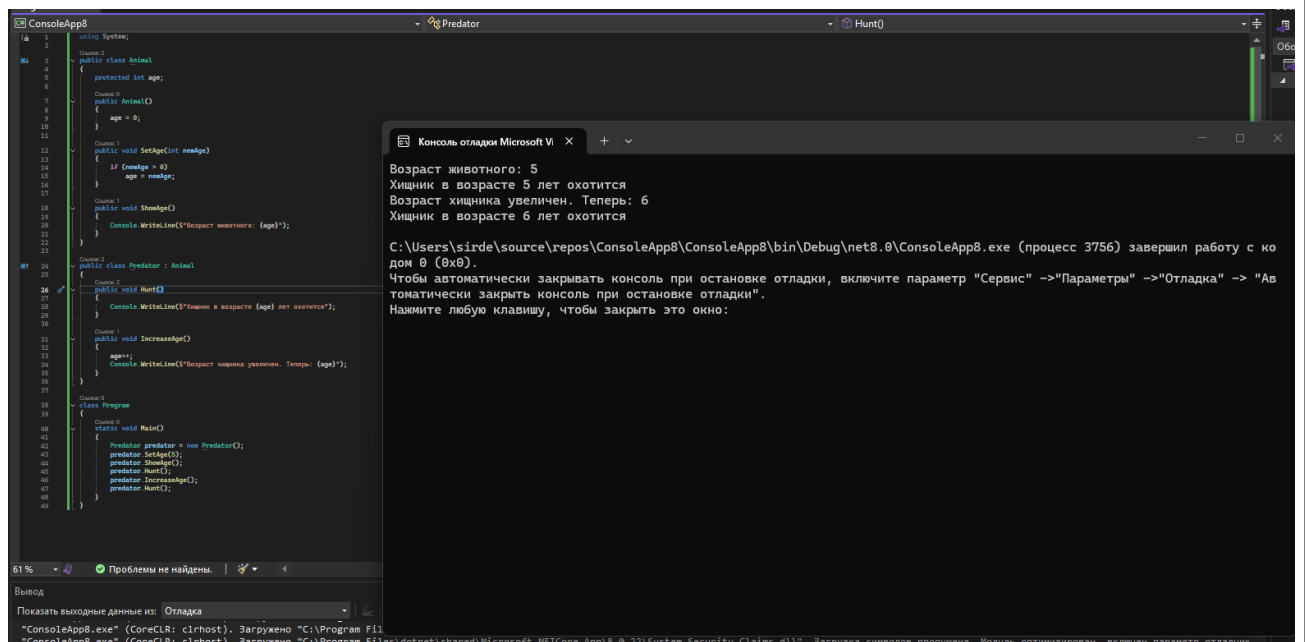


## 4.5

## Отчет по дисциплине по Основы алгоритмизации и программирования



4.6



4.7

The screenshot shows a Visual Studio IDE with a C# console application named 'SpaceShuttle'. The code is as follows:

```

1 using System;
2
3 public class Transport
4 {
5     protected string name;
6
7     public Transport()
8     {
9         name = "Неизвестный транспорт";
10    }
11
12    public void SetName(string newName)
13    {
14        name = newName;
15    }
16
17    public void ShowName()
18    {
19        Console.WriteLine($"Название транспорта: {name}");
20    }
21
22
23    public class SpaceShuttle : Transport
24    {
25        public void Launch()
26        {
27            Console.WriteLine($"Космический корабль '{name}' запущен в космос");
28        }
29
30        public void SetShuttleName(string shuttleName)
31        {
32            name = shuttleName;
33            Console.WriteLine($"Корабль присвоено имя: {name}");
34        }
35    }
36
37    class Program
38    {
39        static void Main()
40        {
41            SpaceShuttle shuttle = new SpaceShuttle();
42            shuttle.SetShuttleName("Атлантис");
43            shuttle.ShowName();
44            shuttle.Launch();
45        }
46    }

```

The debug console output is as follows:

```

Корабль присвоено имя: Атлантис
Название транспорта: Атлантис
Космический корабль 'Атлантис' запущен в космос

C:\Users\sirde\source\repos\ConsoleApp8\ConsoleApp8\bin\Debug\net8.0\ConsoleApp8.exe (процесс 1524) завершил
работу с кодом 0 (0x0).
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отл
адка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно:

```

The status bar at the bottom indicates that the application is running on the CLR host and is loaded from the local file system.

## 5 Полиморфизм

5. Используя , напишите программный код следующих заданий.

5.1. Создайте класс Strategy с виртуальным методом

Display. Метод Display распечатывает на консольном окне

сообщение «Strategy».

5.2. Создайте класс Weather с виртуальным методом Show.

Метод Show отображает на консольном окне

сообщение

«My

Weather».

Создайте

пляр класса Weather и для созданного экземпляра

класса

вызовите метод Show.

5.3.

Создайте два класса Strategy и ConservativeStrategy. Класс Strategy является базовым, а ConservativeStrategy – дочерним. Реализуйте виртуальный метод Display в классе Strategy и переопределите его в дочернем классе ConservativeStrategy.

5.4.

Пусть Animal является базовым классом, а Cat – дочерним.

В классе Animal создайте закрытое строковое поле type, а в

классе Cat закрытое целочисленное поле age. Для классов

Animal, Cat реализуйте стандартные конструкторы. В

стандартном конструкторе класса Animal полю type присваивается значение “My Type”, а в стандартном

конструкторе класса Cat поле age инициализируется значением 5. Создайте виртуальный метод Print в классе Animal.

Методом

Print

в классе Animal распечатывается значение поля type, а в классе Cat распечатывается значение поля age.

Создайте по одному объекту классов Animal, Cat. Для

каждого объекта вызовите метод Print.

5.5. Пусть Entity является абстрактным классом. В классе

Entity реализуйте абстрактный метод Display.

5.6.

Пусть Entity является абстрактным базовым классом, а Product – производным классом. Абстрактный метод

Display объявлен в классе Entity, а в классе Product этот

метод пере- определен для отображения в консольном

окне сообщения «My Product». Создайте объект класса Pr

oduct и для этого объекта вызовите метод Display.

5.7. Создайте интерфейс IPrintable. В этом интерфейсе

определите метод Display без параметров. Затем

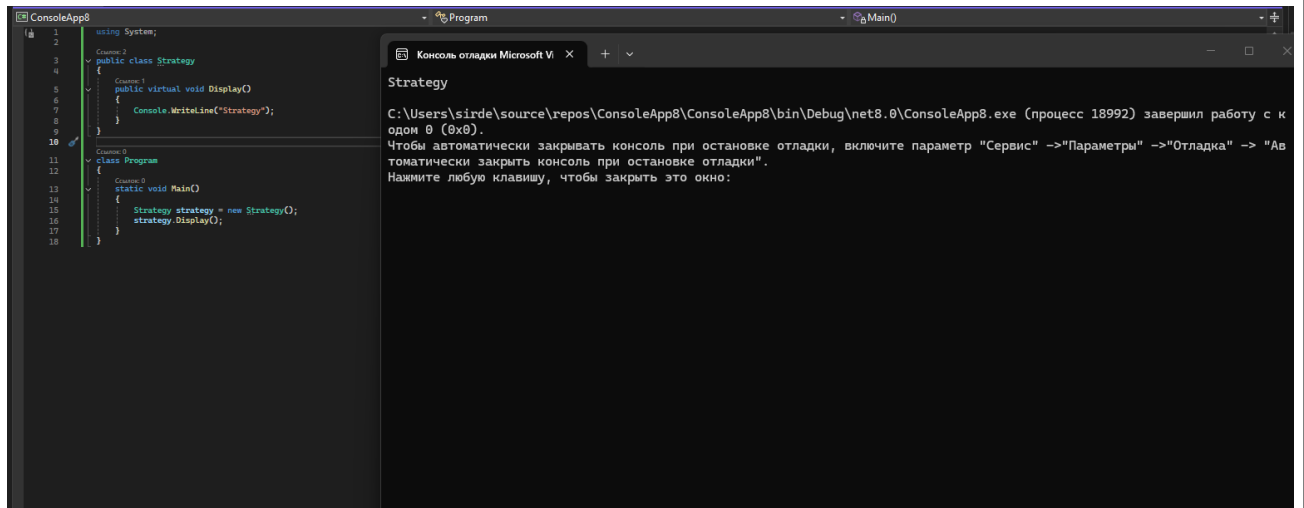
определите класс ConsolePrinting с реализацией методов

интерфейса IPrintable для печати сообщения «My Console»

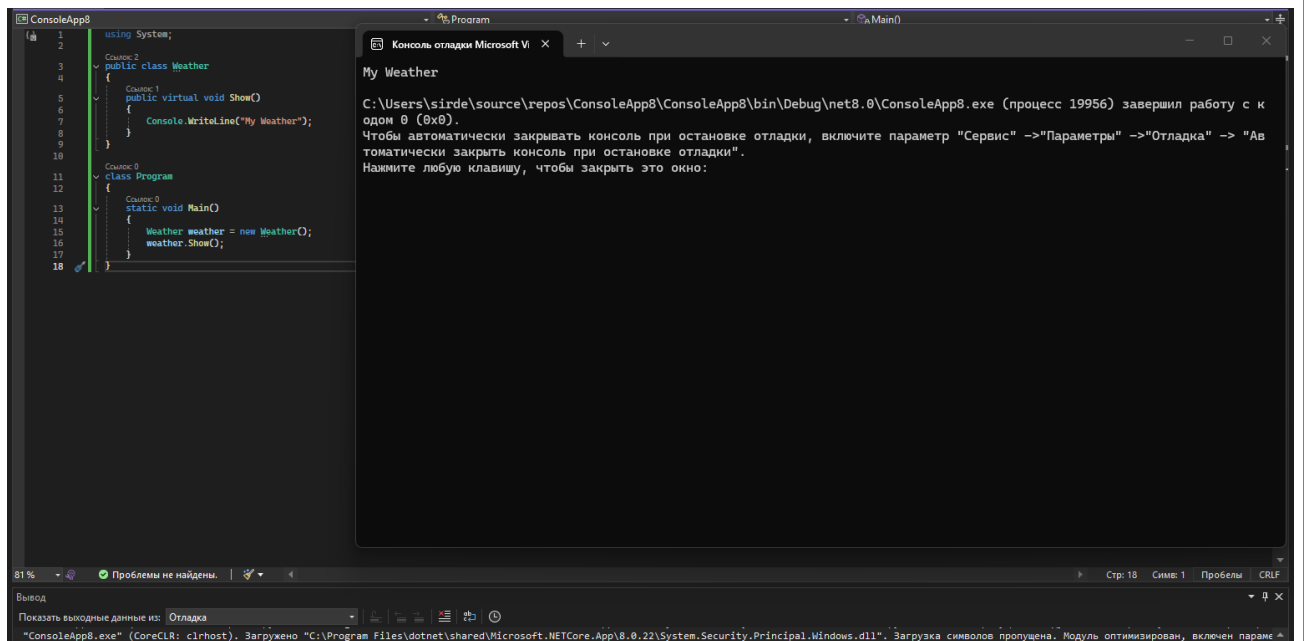
на консольном окне. Создайте объект класса ConsolePrinting

и вызовите для этого объекта метод Display.

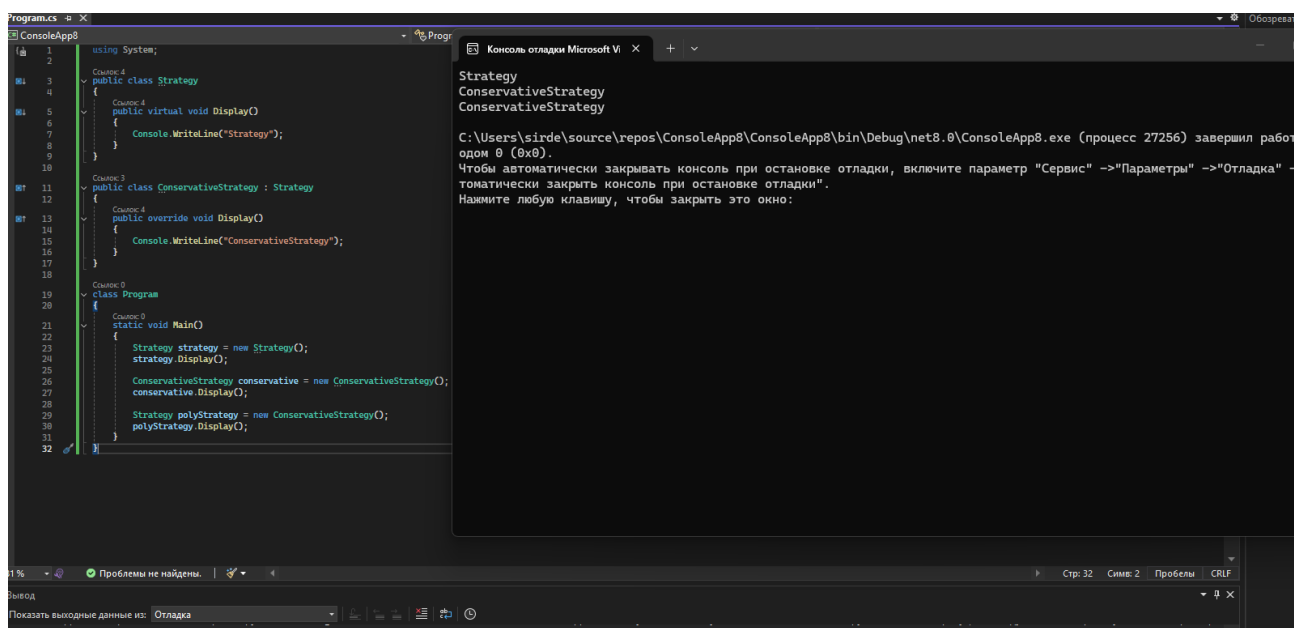
## 5.1



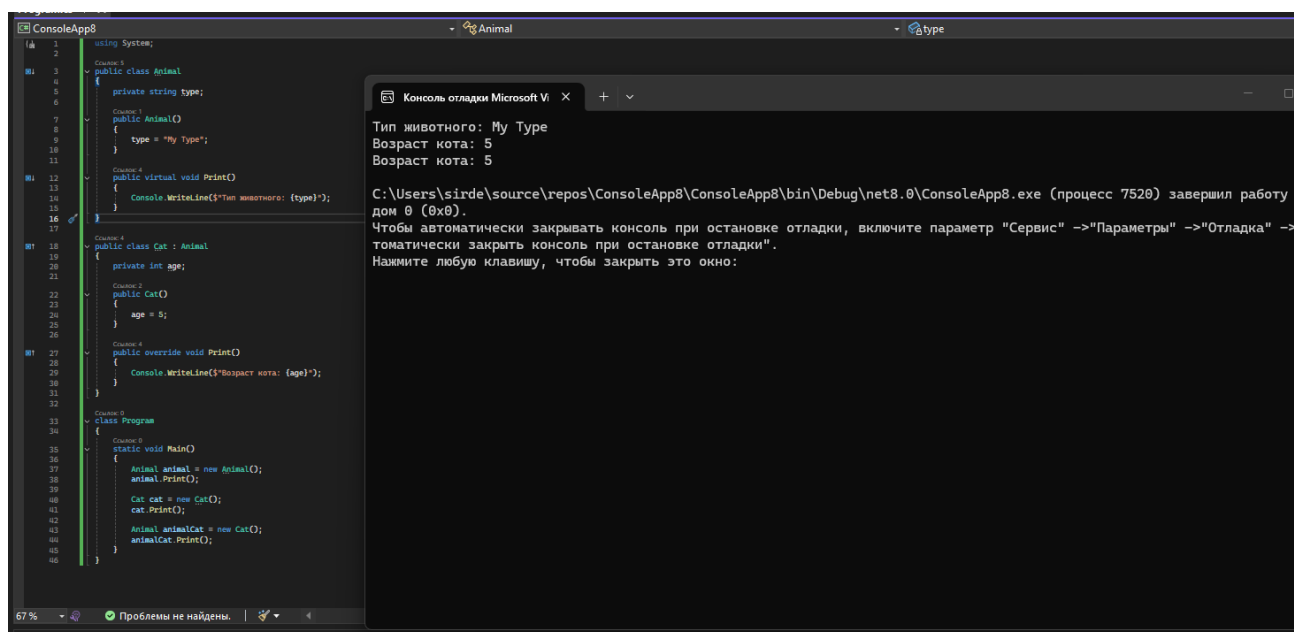
## 5.2



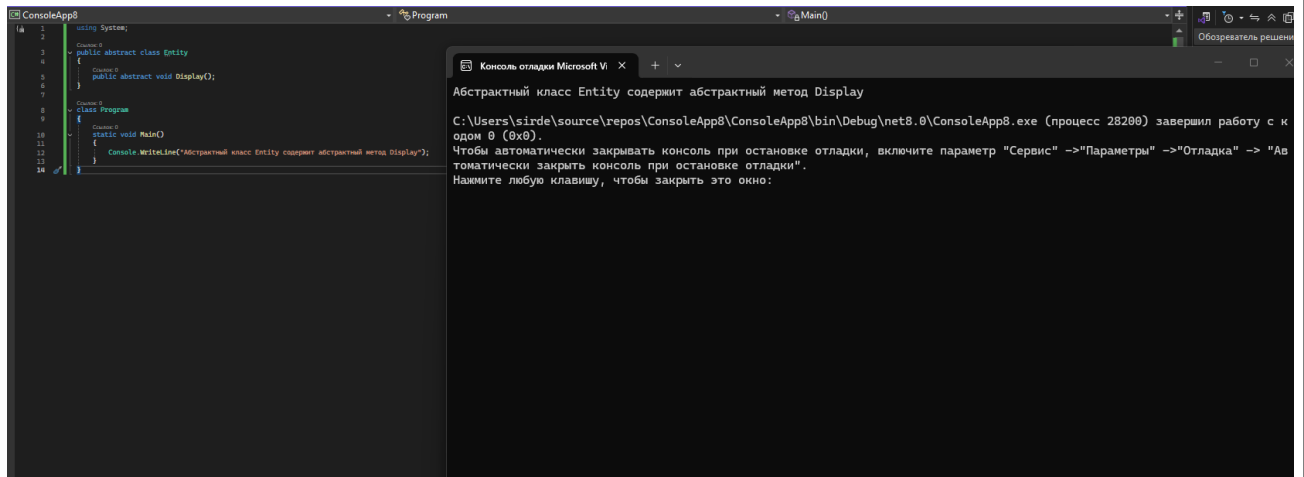
## 5.3



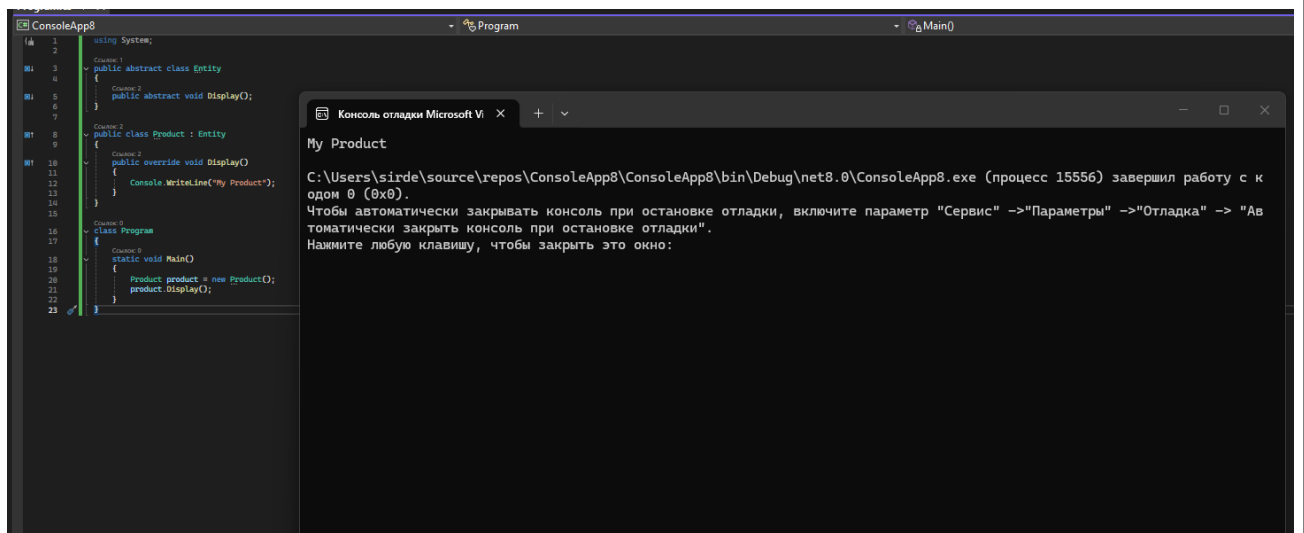
## 5.4



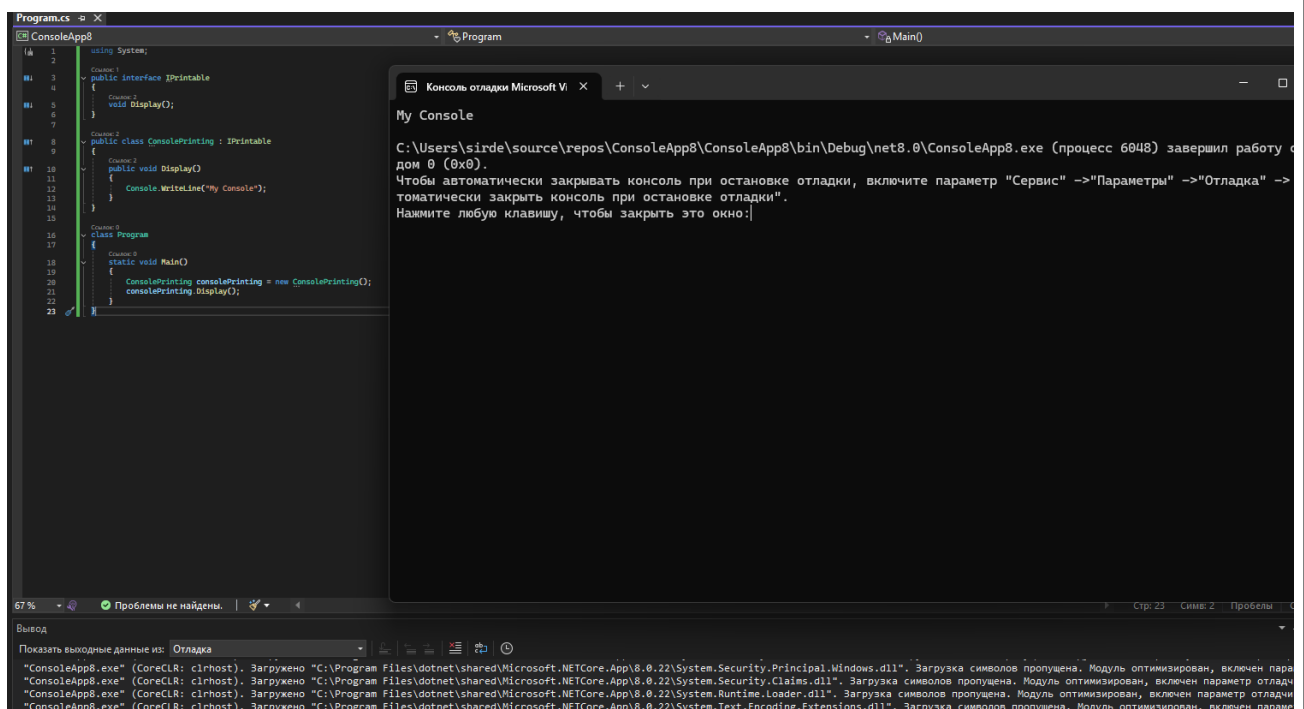
## 5.5



## 5.6



## 5.7



### **Официальная литература:**

1 Microsoft, Learn. Документация по C#. Вложенные типы. —

2024

— URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/nested-types> (дата обр. 27.02.2024).

2 Microsoft, Learn. Документация по C#. Конструкторы.

— 2024

— URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/classes-and-structs/constructors> (дата обр. 27.02.2024).

3 Microsoft, Learn. Документация по C#. Наследование.

— 2024

— URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/fundamentals/object-oriented/inheritance> (дата обр. 27.02.2024).

4 Microsoft, Learn. Документация по C#. Полиморфизм.

— 2024

— URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/fundamentals/object-oriented/polymorphism> (дата обр. 27.02.2024).

5 Microsoft, Learn. Документация по C#. Свойства. — 2024 —

URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/properties> (дата обр. 27.02.2024).

6 Microsoft, Learn. Документация по C#. Стандартные блоки про-

граммы. — 2024 — URL: <https://docs.microsoft.com>

/ ru - ru / dotnet / csharp / tour -

of -

csharp / program -

building -

27.02.2024).

7 Microsoft, Learn. Документация по C#. Типы и члены.

— 2024

— URL: [https : / / docs . microsoft . com / ru - ru / dotn](https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/types)

[et / csharp / tour - of - csharp/types](https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/types) (дата обр.

27.02.2024).

**Практическая работа подключение к MySQL  
server .docx**

