МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное образовательное учреждение высшего образования

САНКТ – ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | Д.О. Шевяков |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7  (ВАРИАНТ №27)  ПОЛИМОРФИЗМ. ВИРТУАЛЬНЫЕ МЕТОДЫ КЛАССОВ. |
|  |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4321 |  |  |  | П.А. Евстафьева |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

**СОДЕРЖАНИЕ**

[1 Постановка задачи 3](#_gjdgxs)

[2 Выполнение задачи](#_30j0zll) 3

[2.1 Описание реализованной функции](#_3znysh7) 3

[2.2 Листинг программы](#_2et92p0) 4

[2.3 Тесты работы программы](#_tyjcwt) 6

2.4 Интерфейс программы 7

[3 Выводы](#_3dy6vkm) 7

**1 Постановка задачи**

Необходимо разработать Windows Forms приложение на языке C# для управления списком самолётов. Программа должна поддерживать следующие функции:

Разработать собственный вариант переопределения метода базового класса, протестируйте варианты.

Таблица 1 – Индивидуальное задание по варианту №27

|  |  |  |
| --- | --- | --- |
| Задание | Вход | Выход |
| **Данные:**   * Модель самолета; * Авиакомпания; * Дальность полета самолета; * Запас горючего.   **Задача:**  Провести сортировку самолетов компании по дальности полета. |  |  |

**2 Выполнение задачи**

**2.1 Описание реализованной функции**

В представленном коде реализован класс Airplane, который служит базовым для создания различных типов самолетов, таких как PassengerAirplane и CargoAirplane. Этот класс включает свойства, такие как Name, Model, Range, FuelConsumption, ManufactureDate и Foto, которые описывают основные характеристики самолета.

Метод GetInfo() предоставляет строковое представление информации о самолете, включая его имя, модель, дальность полета, расход топлива и дату производства. Этот метод переопределяется в дочерних классах, чтобы дополнительно выводить специфическую информацию, например, вместимость пассажиров и наличие бизнес-класса для пассажирских самолетов, а также грузоподъемность и тип груза для грузовых самолетов.

Класс также включает статические методы WriteToFile и ReadFromFile, которые позволяют сохранять и загружать список самолетов в текстовый файл. Метод WriteToFile перебирает список самолетов и записывает их свойства в файл в формате CSV, что упрощает дальнейшую обработку данных. Метод ReadFromFile считывает данные из файла, создает объекты Airplane и добавляет их в список, проверяя, чтобы все поля были заполнены корректно.

**2.2 Листинг программы**

Класс Airplane на рисунках 1,2:

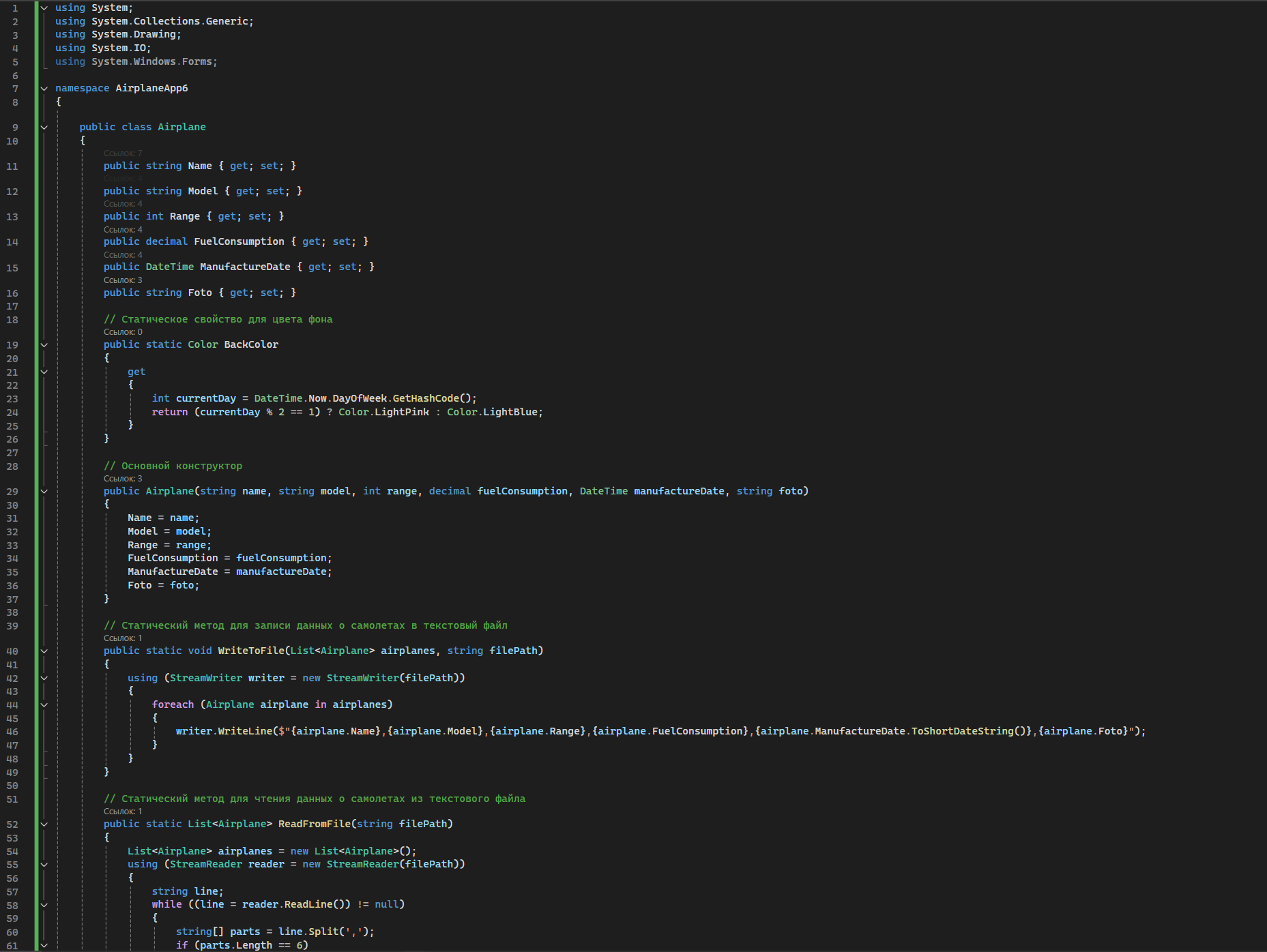


Рисунок 1 – Код класса Airplane

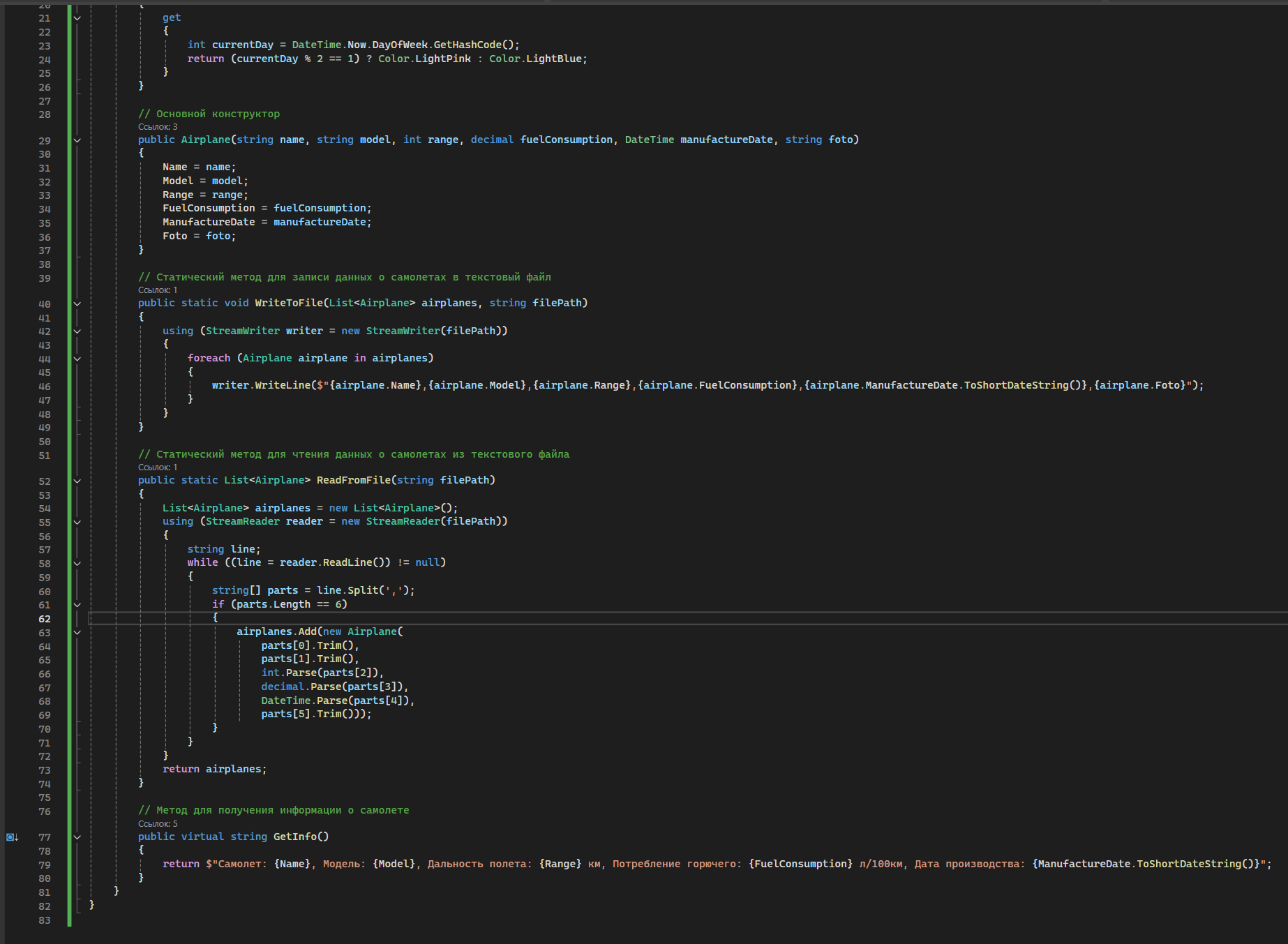


Рисунок 2 – Код класса Airplane

Код формы (Form1) на рисунках 3, 4, 5:

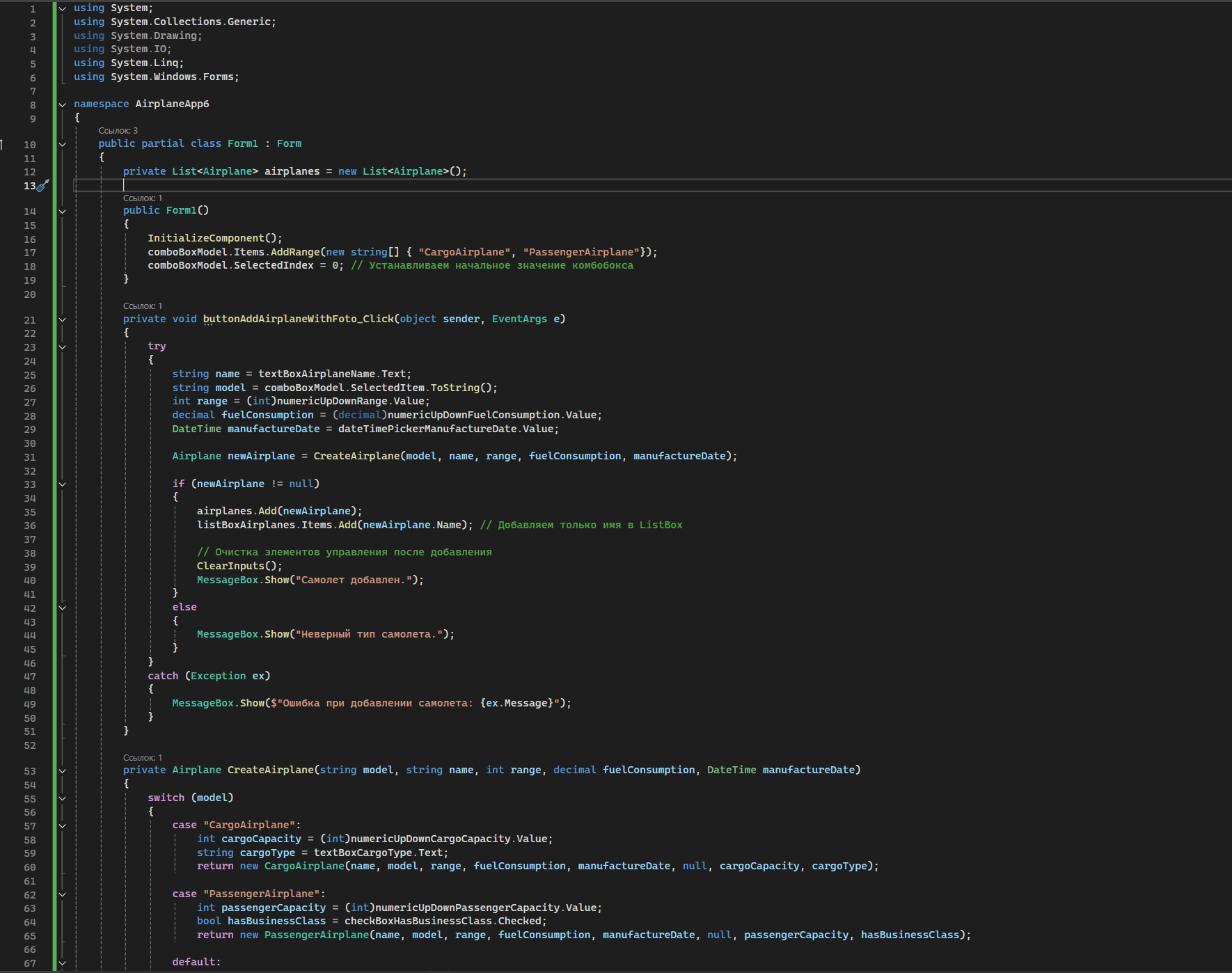


Рисунок 3 – Код формы (Form1)

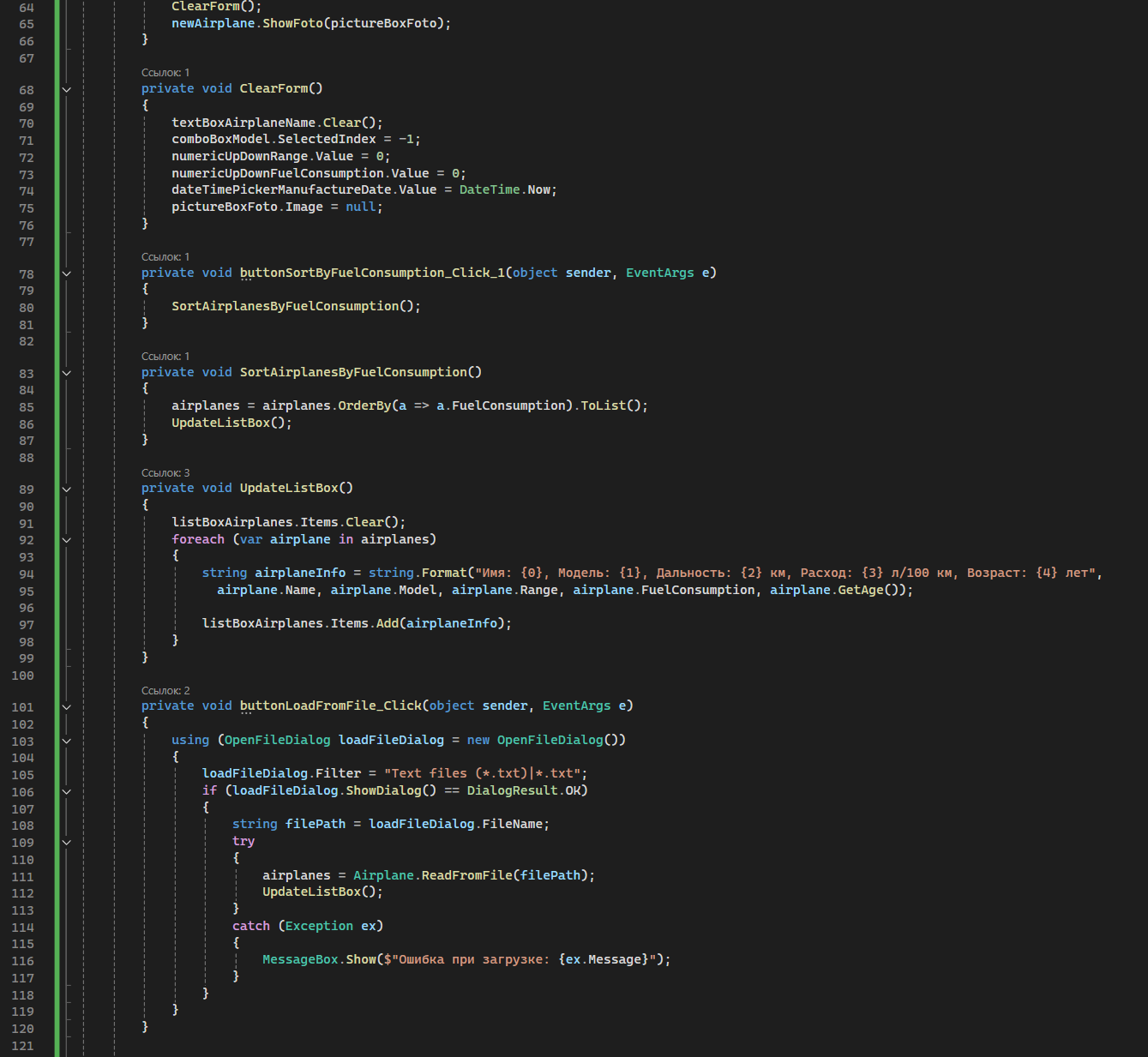
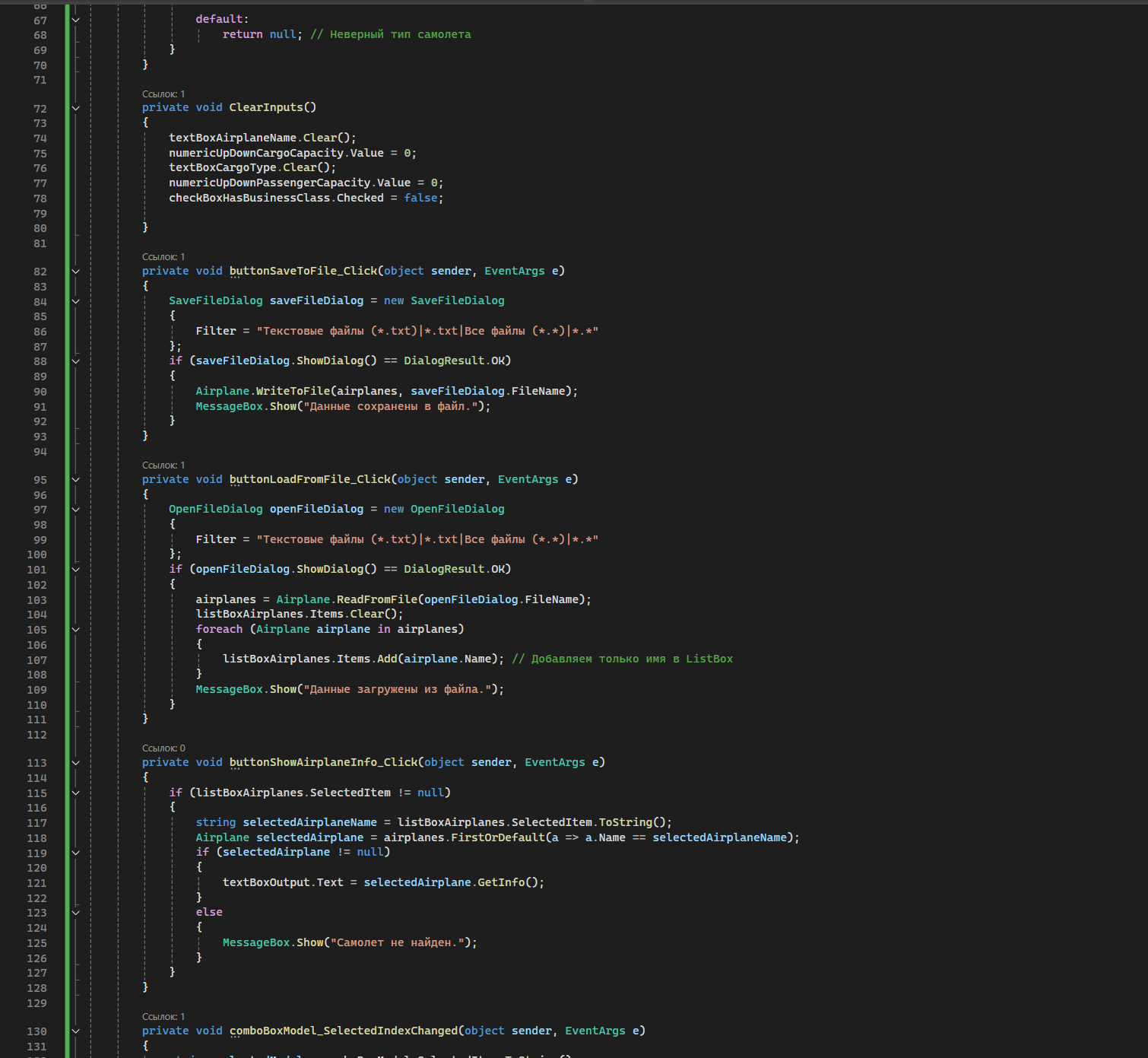
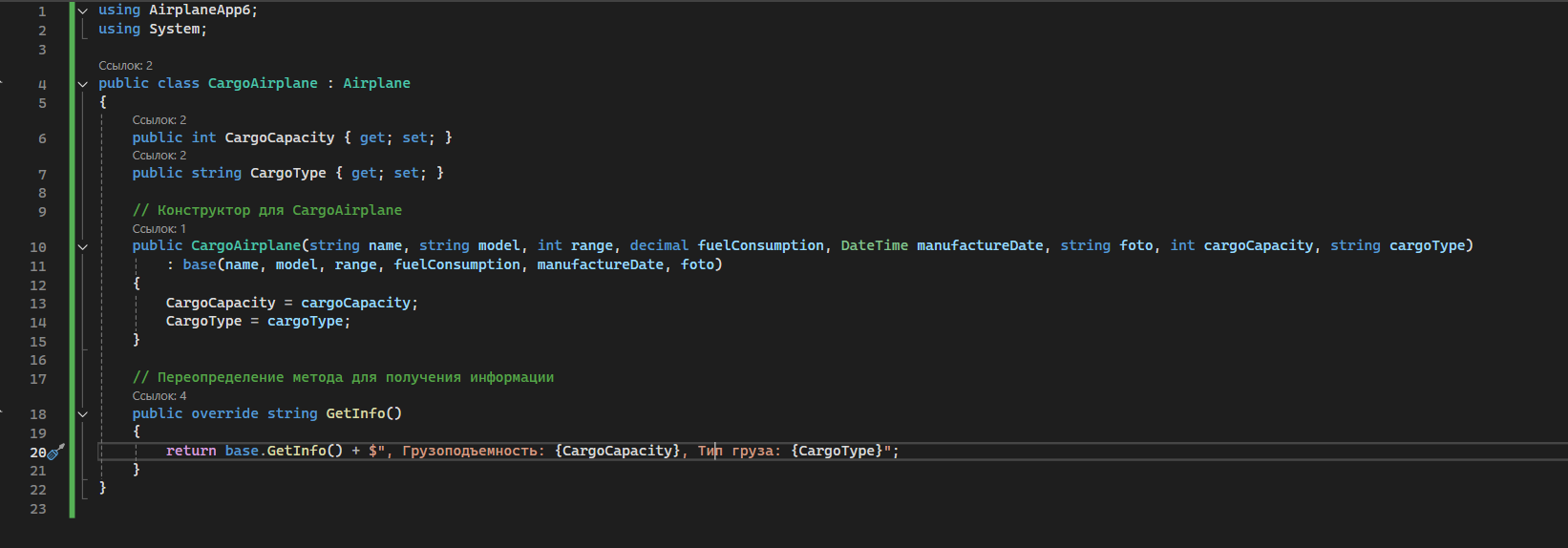


Рисунок 4 – Код формы (Form1)



Рисунок 5 – Код формы (Form1)



На рисунке 7 изображен код дочернего класса CargoAirplane:

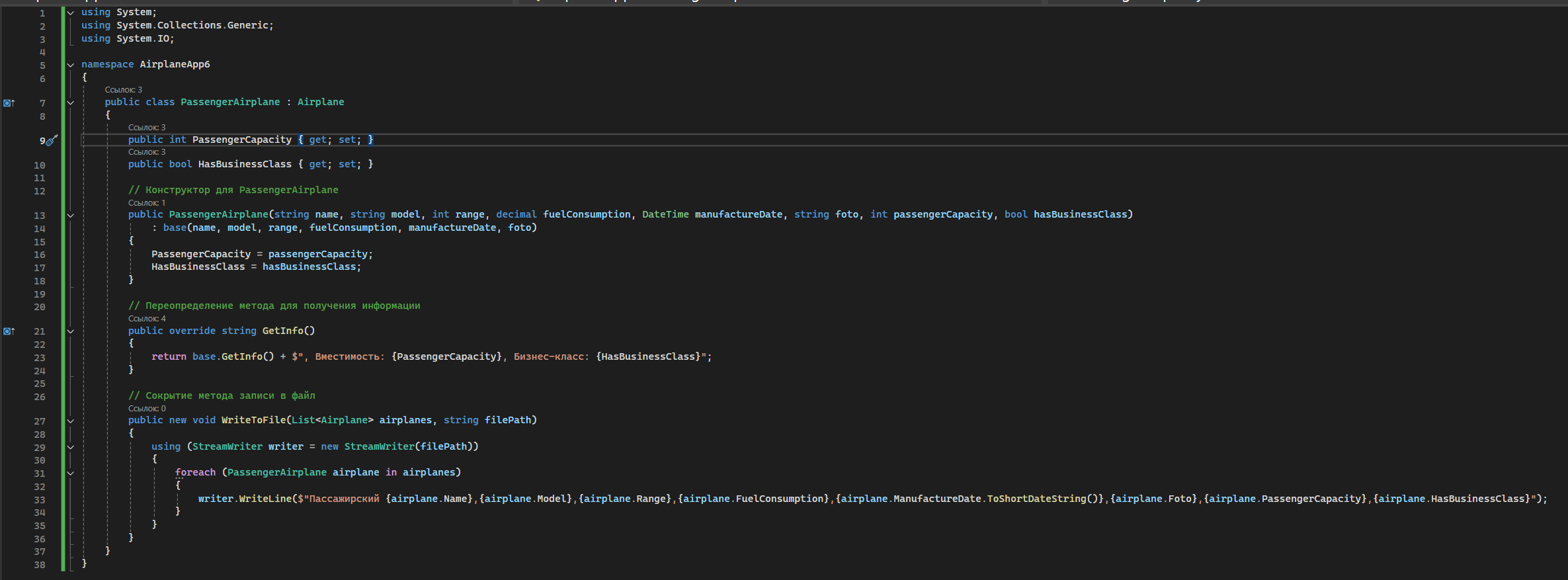


Рисунок 8 – Код дочернего класса PassengerAirplane

Исследование возможности запрета на переопределение в конкретном примере для данного кода на рисунке 9.

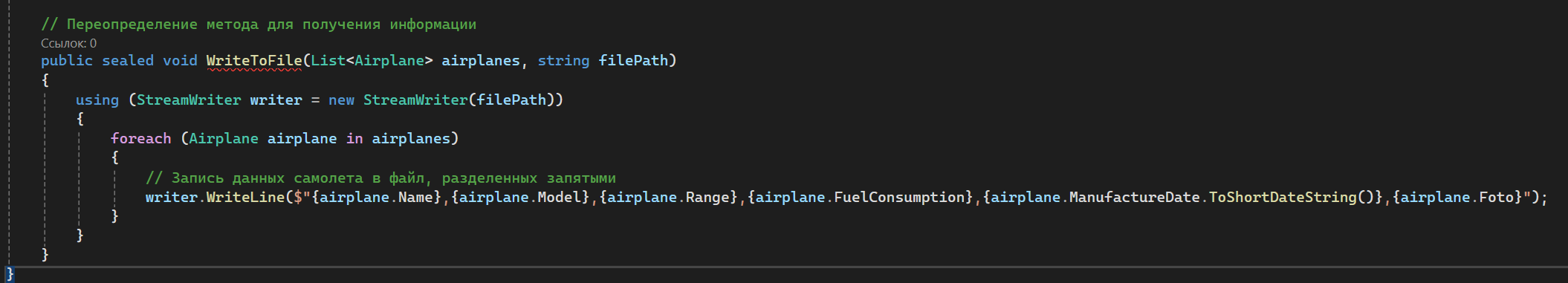


Рисунок 9 – Исследование возможности запрета на переопределение в конкретном примере для данного кода

**2.3 Тесты работы программы**

Программа была протестирована на следующих сценариях:

1. Добавление данных о различных самолетах.

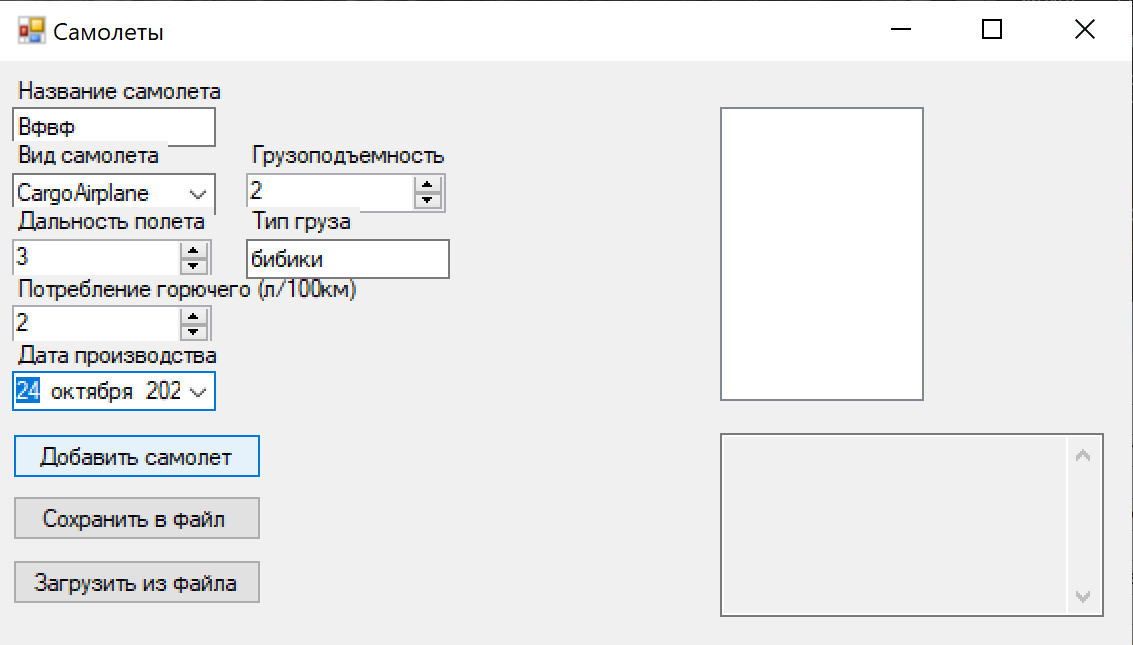


Рисунок 6 – Тест программы

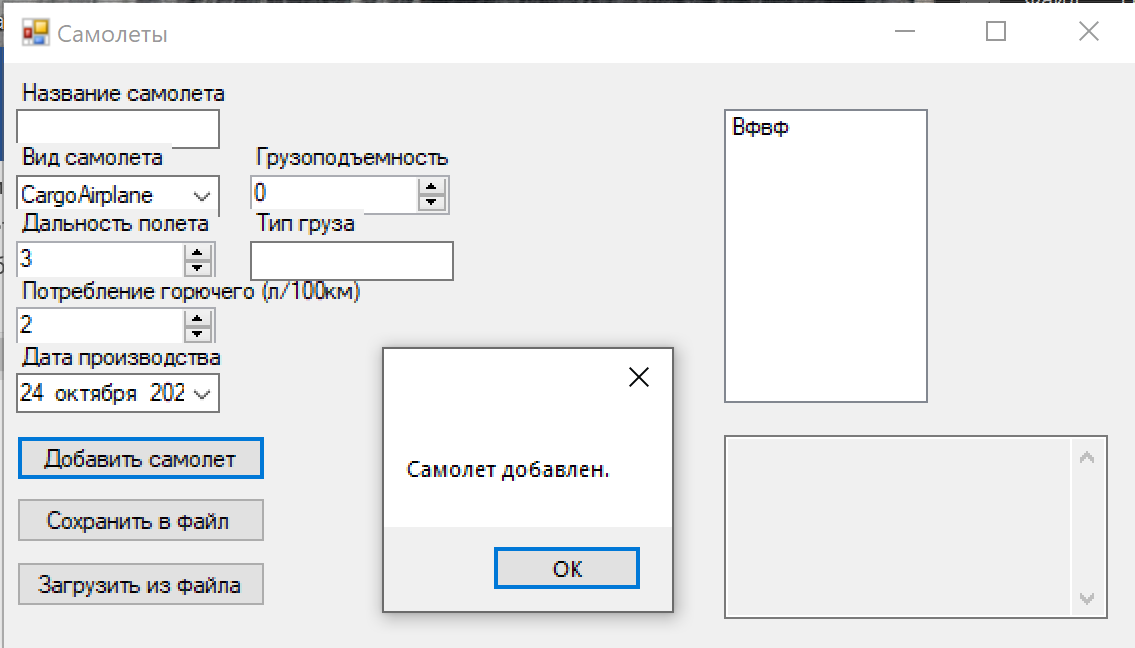


Рисунок 7 – Тест программы

* 1. **Интерфейс программы**

Интерфейс программы AirplaneApp6 представляет собой графическое пользовательское приложение, разработанное с использованием Windows Forms, обеспечивающее интуитивно понятный способ взаимодействия пользователя с данными о самолетах. Основное окно (Form1) содержит несколько ключевых компонентов: TextBox для ввода названия нового самолета, ComboBox для выбора модели с доступными вариантами "CargoAirplane" и "PassengerAirplane", а также NumericUpDown элементы для указания дальности полета и расхода топлива. Пользователь может выбрать дату производства с помощью DateTimePicker, а для грузовых самолетов предусмотрены дополнительные поля для ввода грузоподъемности и типа груза. Для пассажирских самолетов есть NumericUpDown для количества мест и CheckBox для указания наличия бизнес-класса. Список добавленных самолетов отображается в ListBox, где пользователь может выбирать элементы для просмотра информации о каждом самолете в TextBox. В интерфейсе предусмотрены кнопки: "Добавить самолет", "Сохранить в файл", "Загрузить из файла" и "Показать информацию о самолете". Каждая кнопка выполняет свои функции, такие как добавление нового самолета с проверкой данных, сохранение информации в текстовый файл и загрузка данных из файла. Динамическое изменение интерфейса осуществляется с помощью метода ToggleVisibility, который управляет видимостью элементов управления в зависимости от выбранной модели самолета. Таким образом, интерфейс AirplaneApp6 спроектирован для удобства пользователя, позволяя легко вводить, редактировать и сохранять данные о самолетах, а также адаптироваться под потребности пользователя в зависимости от выбранного типа самолета.

**3 Выводы**

В ходе лабораторной работы было создано Windows Forms приложение, которое реализует функции управления списком самолётов, их сортировки и отображения. Программа демонстрирует основные принципы объектно-ориентированного программирования, включая использование классов, перегрузку конструкторов, а также использование структур для хранения данных.