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 №8  (ВАРИАНТ №27)  ОТНОШЕНИЕ МЕЖДУ КЛАССАМИ. |
|  |
| 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 и описать ассоциацию, агрегацию и композицию между классами.

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**

Основной задачей программы является добавление самолетов в список, просмотр информации о каждом из них, а также выполнение операций, таких как заправка самолетов топливом. Программа предоставляет пользователю возможность через графический интерфейс взаимодействовать с объектами самолетов, добавлять новые самолеты, просматривать их характеристики и выполнять операции, такие как заправка топливом для выбранных самолетов.

Для реализации этого функционала в программе используется интерфейс IAircraft, который служит общей абстракцией для различных типов самолетов, таких как грузовые (CargoAirplane) и пассажирские (PassengerAirplane). Интерфейс включает в себя базовые свойства, такие как имя самолета, модель, дальность полета и расход топлива, а также метод GetInfo(), который возвращает строковое представление информации о самолете. Это позволяет работать с различными типами самолетов единообразно, не завися от конкретной реализации.

Кроме того, в проекте реализован интерфейс IFuelCapacity, который добавляет возможность заправки самолетов топливом. Этот интерфейс включает свойство FuelConsumption (расход топлива) и метод Refuel(), который позволяет увеличивать расход топлива при заправке самолета. Классы CargoAirplane и PassengerAirplane реализуют оба интерфейса, что позволяет выполнять операции как для грузовых, так и для пассажирских самолетов через единый интерфейс.

1. **Ассоциация**: Ассоциация представляет собой связь между объектами, где каждый объект может существовать независимо от другого. В нашем проекте ассоциация проявляется между классом формы (Form1) и объектами самолетов. Форма управляет списком самолетов (List<IAircraft>), что означает, что форма и самолеты находятся в ассоциативных отношениях. Самолеты добавляются, отображаются и обновляются в списке формы, но форма не является владельцем самолетов. Самолеты могут существовать независимо от формы.
2. **Агрегация**: Агрегация представляет собой более слабое отношение, где один объект может быть частью другого, но может существовать отдельно. В данном случае, самолеты являются частью агрегированного списка самолетов, который управляется классом формы (Form1). Если форма будет уничтожена, то список самолетов также исчезнет, однако сами объекты самолетов могут существовать и без формы. Это означает, что самолеты можно рассматривать как агрегированные объекты в коллекции.
3. **Композиция**: Композиция — это более сильная форма агрегации, когда один объект полностью зависит от другого. В нашем проекте композиция может проявляться в том, как создаются экземпляры классов самолетов. Например, каждый самолет имеет свои параметры, такие как Foto, которые создаются и управляются внутри класса самолета. Если сам самолет уничтожается, его компоненты также исчезают. В данном случае компоненты (например, изображения или детали самолета) могут быть частью объекта самолета и не существовать отдельно без него.

**2.2 Листинг программы**

Класс Airplane на рисунке 1:

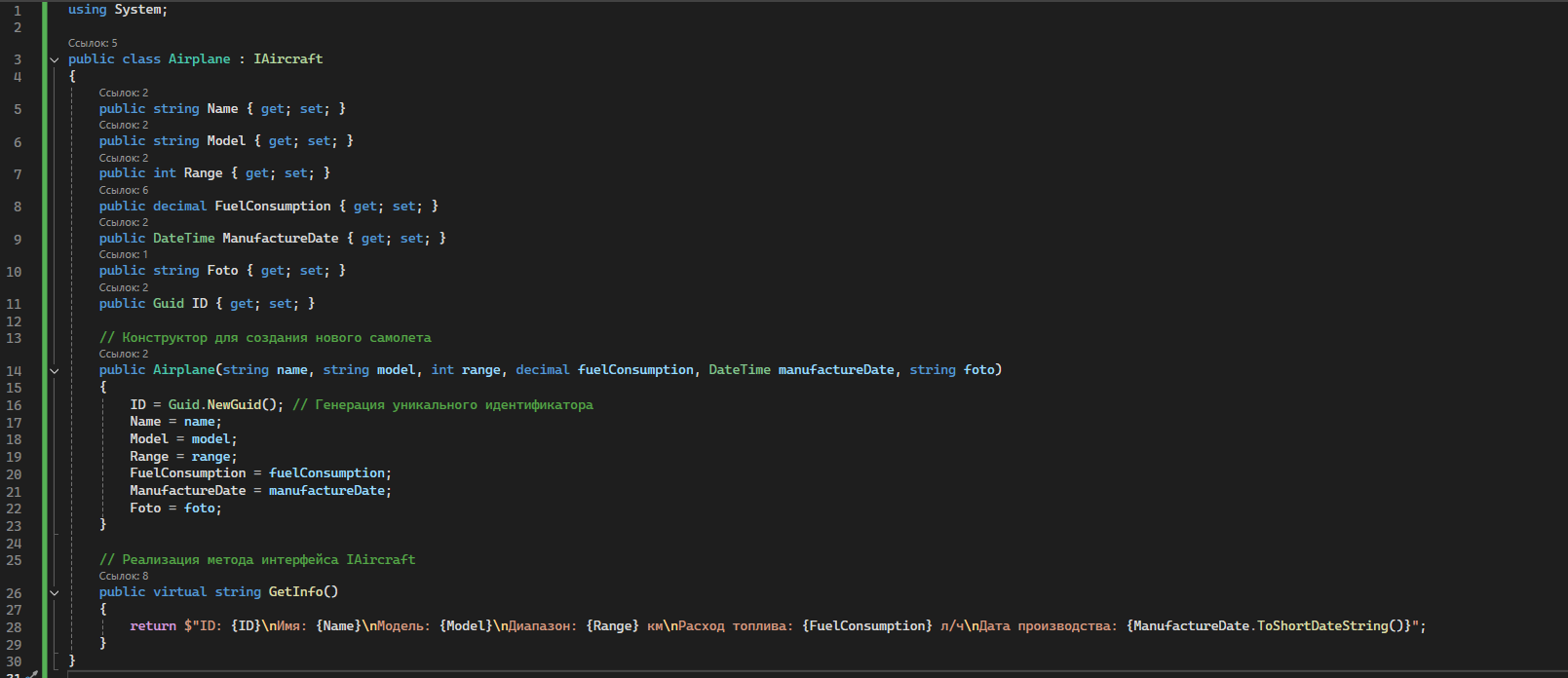


Рисунок 1 – Код класса Airplane

На рисункфх 2, 3 изображен код интерфейсов:

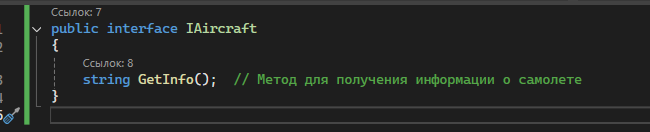


Рисунок 2 – Код интерфейса IAircraft

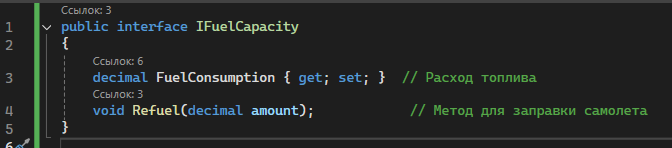


Рисунок 3 – Код интерфейса IFuelCapacity

Код формы (Form1) на рисунках 4, 5, 6:

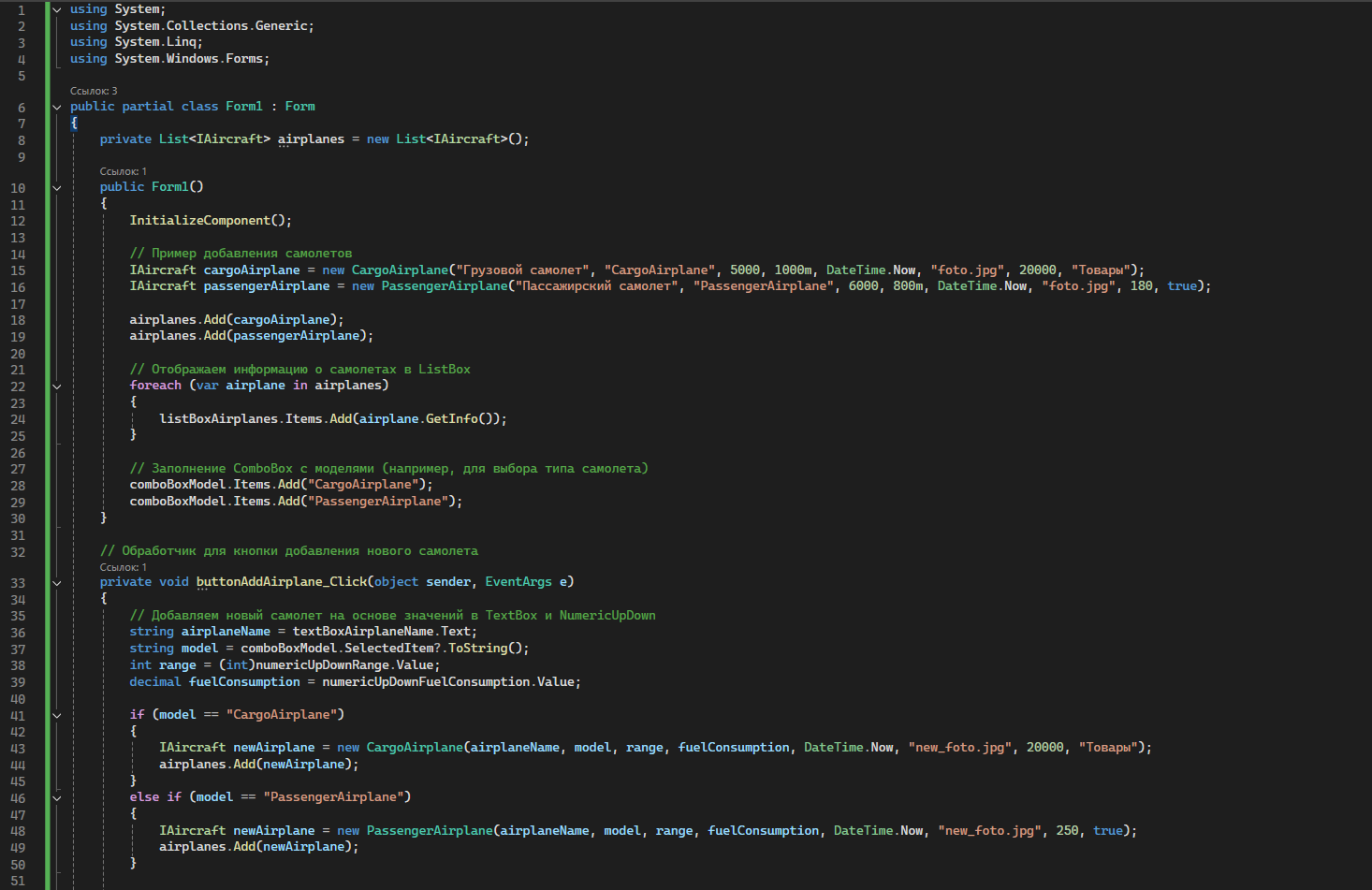


Рисунок 4 – Код формы (Form1)

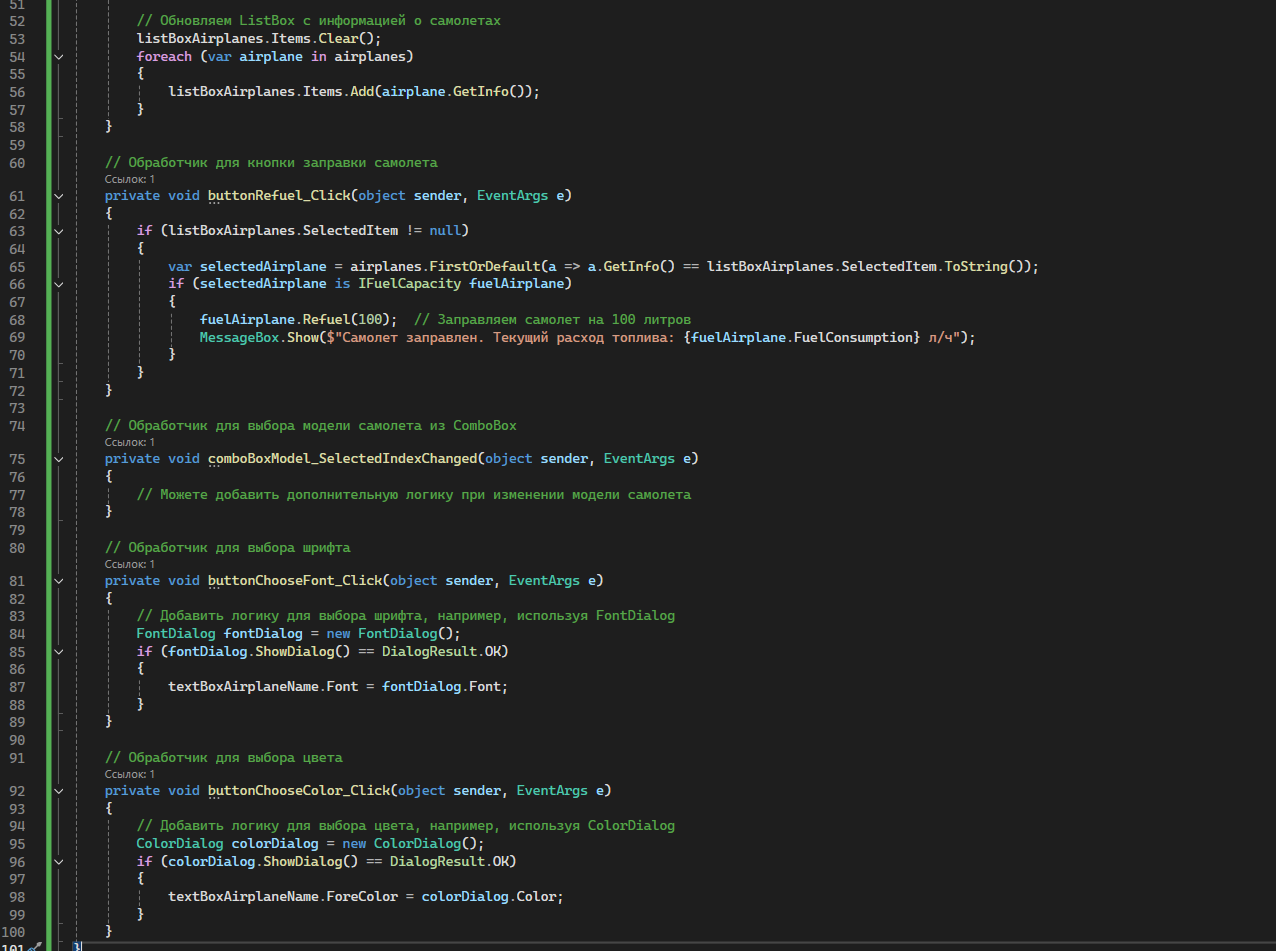


Рисунок 5 – Код формы (Form1)

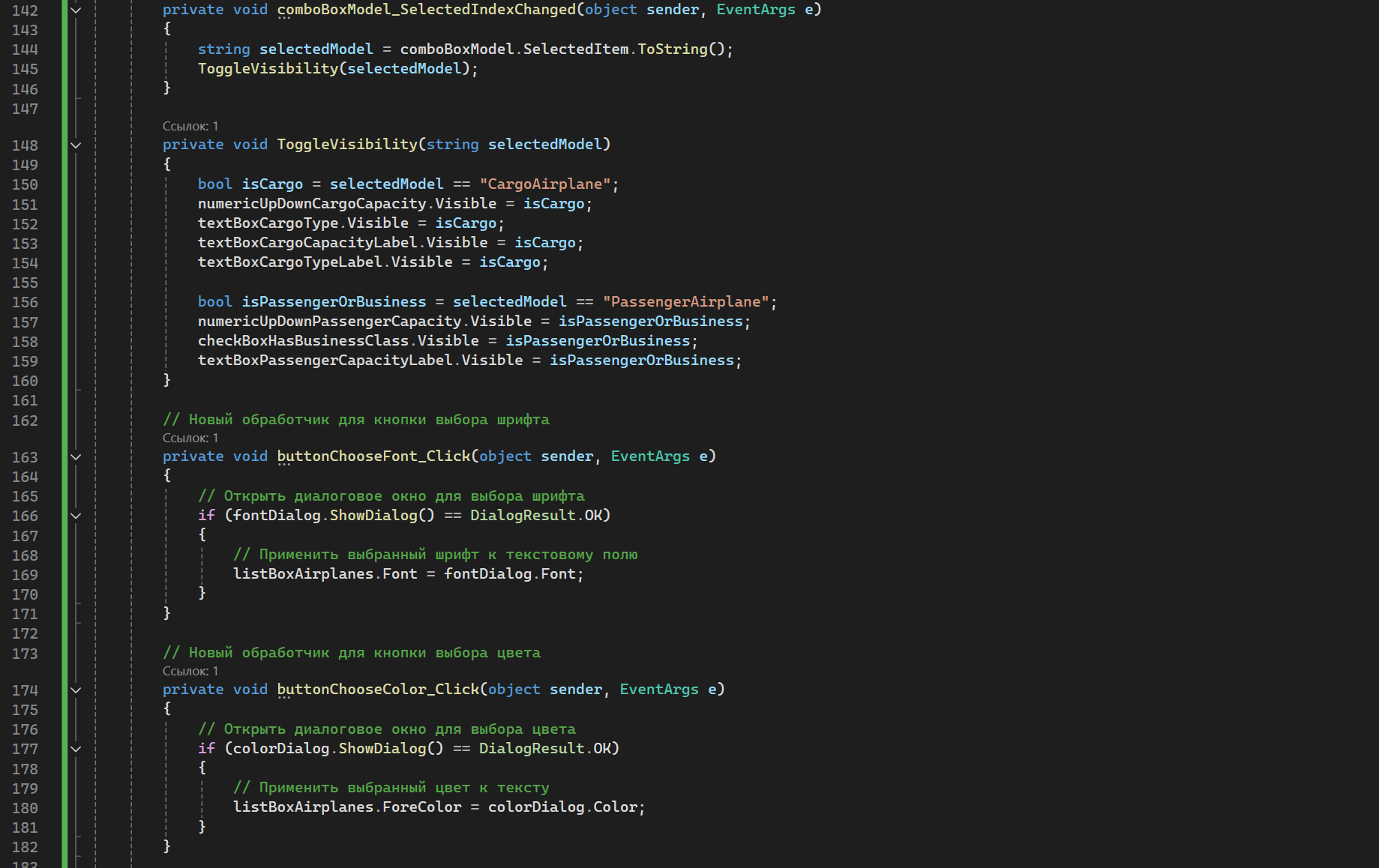


Рисунок 6 – Код формы (Form1)

На рисунке 7 изображен код дочернего класса CargoAirplane:

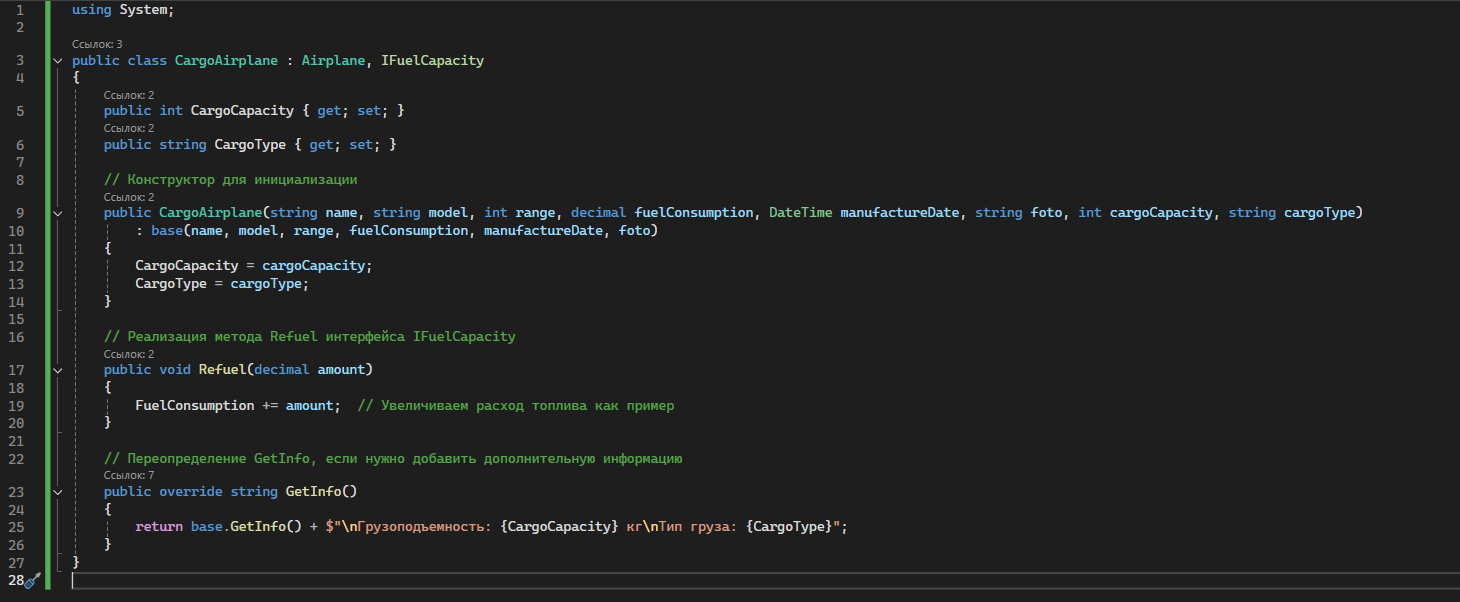


Рисунок 7 – Код дочернего класса CargoAirplane

На рисунке 8 изображен код дочернего класса PassengerAirplane:

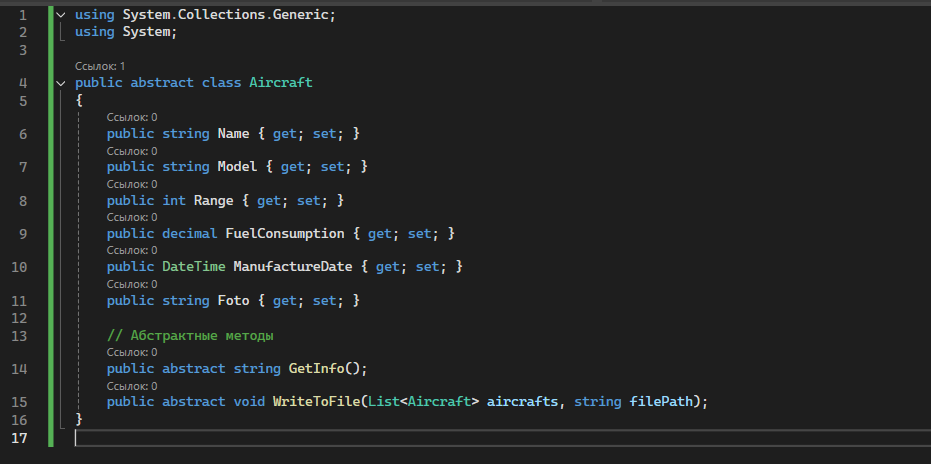


Рисунок 8 – Код дочернего класса PassengerAirplane

На рисунке 9 изображен код абстрактного класса Aircraft1:

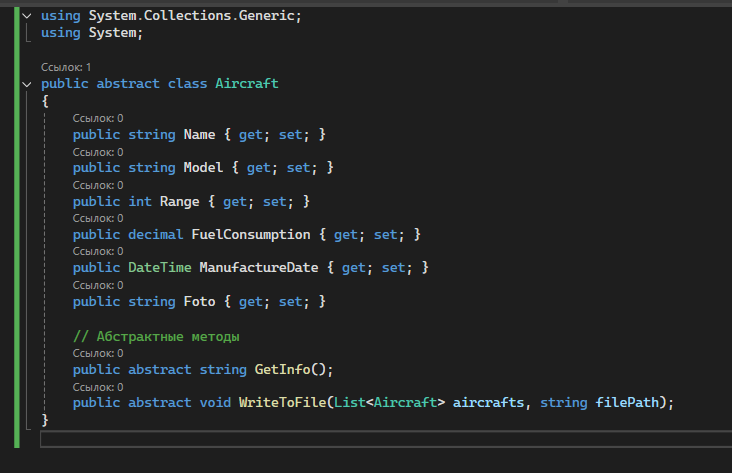


Рисунок 9 – Код абстрактного класса Aircraft1

На рисунке 10 изображен код класса с агригацией:

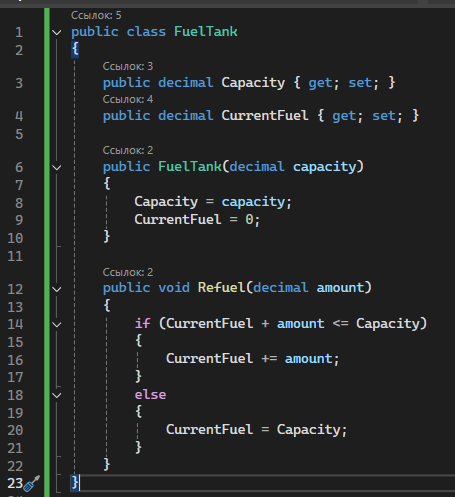


Рисунок 10 – Код класса FuelTank

На рисунке 11 изображен код класса Wing:

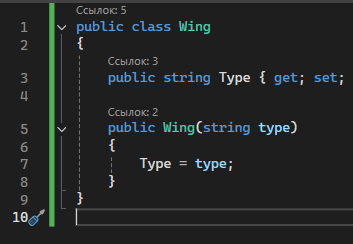


Рисунок 11 – Код класса Wing

**2.3 Тесты работы программы**

Программа была протестирована на следующих сценариях:

1. Заправка самолета

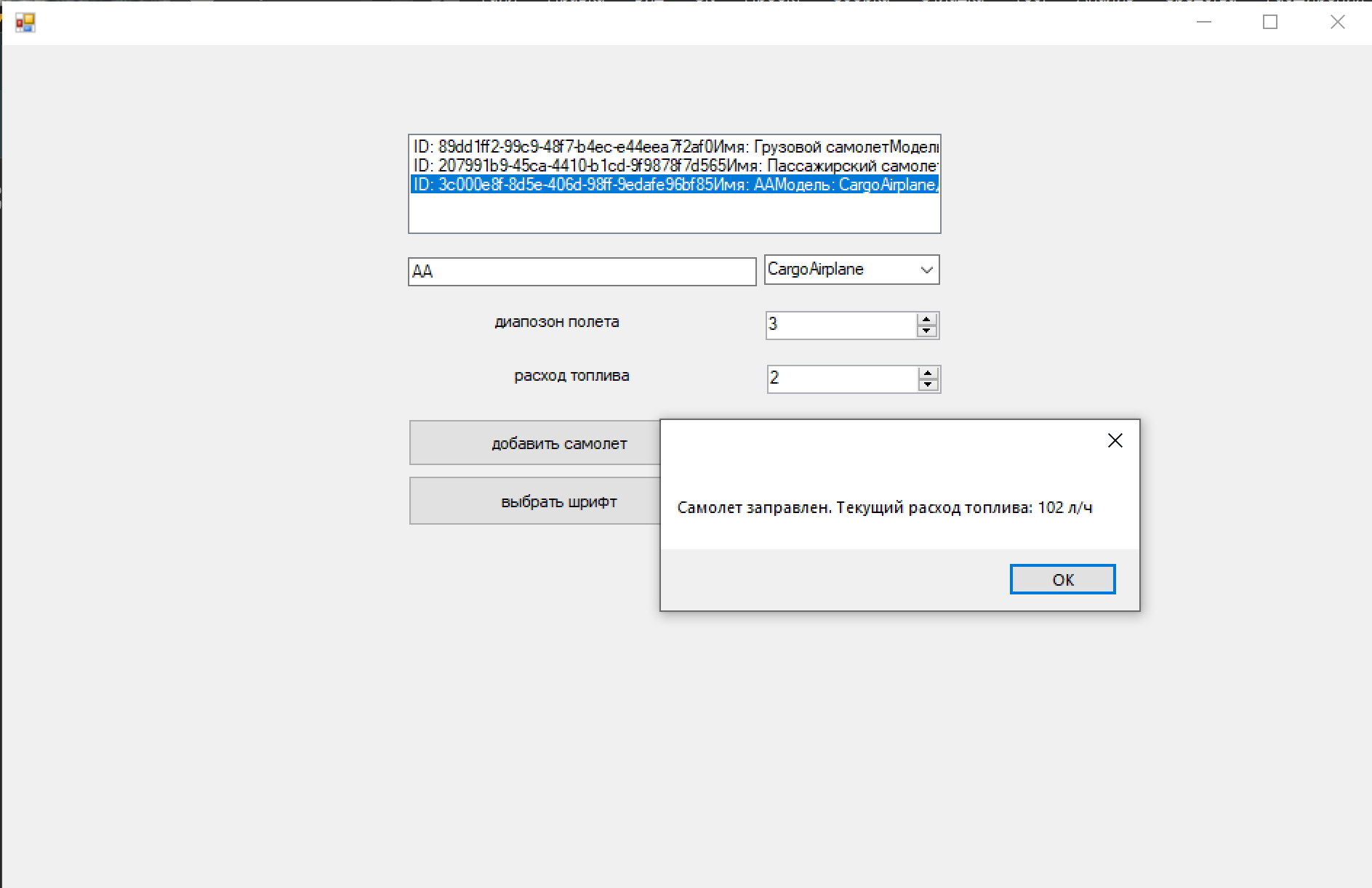


Рисунок 12 – Результат работы программы

* 1. **Интерфейс программы**

Интерфейс программы представляет собой графическое окно с элементами управления, которые позволяют пользователю взаимодействовать с функциональностью приложения для работы с самолетами. Основные компоненты интерфейса включают в себя:

1. **Список самолетов** — ListBox (listBoxAirplanes), который отображает все добавленные самолеты. Пользователь может выбрать самолет из этого списка, чтобы просмотреть информацию о нем или выполнить действия, такие как заправка. Этот элемент позволяет отображать информацию о самолетах в виде строк, предоставленных методами, реализованными в соответствующих классах самолетов.
2. **Поле для ввода имени самолета** — TextBox (textBoxAirplaneName), где пользователь может ввести название нового самолета. Это поле используется при добавлении нового самолета в список, и помогает идентифицировать самолет.
3. **Выпадающий список для выбора модели самолета** — ComboBox (comboBoxModel), в котором отображаются различные доступные модели самолетов. При изменении выбранной модели в списке активируется соответствующая логика (например, изменение диапазона полета или расхода топлива, в зависимости от выбранной модели).
4. **Поле для ввода диапазона полета** — NumericUpDown (numericUpDownRange), которое позволяет пользователю указать дальность полета нового самолета. Этот элемент управления ограничивает ввод только числами и позволяет установить минимальное и максимальное значения для диапазона полета.
5. **Поле для ввода расхода топлива** — NumericUpDown (numericUpDownFuelConsumption), где пользователь может задать расход топлива для самолета. Также представлен числовой ввод, что дает пользователю возможность задавать конкретные величины расхода топлива.
6. **Кнопка для добавления нового самолета** — Button (buttonAddAirplane), которая запускает процесс добавления нового самолета в список. После нажатия на эту кнопку данные из полей ввода (имя, модель, диапазон полета, расход топлива) используются для создания нового объекта самолета, который затем добавляется в список отображаемых самолетов.
7. **Кнопка для заправки выбранного самолета** — Button (buttonRefuel), которая позволяет пользователю выбрать самолет из списка и заправить его. При нажатии на кнопку для заправки, выбранный самолет заправляется на определенное количество литров топлива (например, 100 литров), и отображается информация о текущем расходе топлива.
8. **Кнопки для изменения шрифта и цвета** — Button (buttonChooseFont и buttonChooseColor), которые могут быть использованы для изменения визуальных параметров интерфейса программы. Эти кнопки позволяют пользователю настроить внешний вид формы, выбрав шрифт или цвет текста.

**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.