

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.02

Выполнил: Епимахов Денис Леонидович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В.

2025

Содержание

[1. **Задание №1 Оконное приложение «Отделы»** 3](#_Toc118960339)

[1.1 Описание задачи 3](#_Toc118960340)

[1.2 Структура проекта 4](#_Toc118960341)

[1.3 Описание разработанных функций 4](#_Toc118960342)

[1.4 Алгоритм решения 8](#_Toc118960343)

[1.5 Используемые библиотеки 9](#_Toc118960344)

[1.6 Тестовые случаи 9](#_Toc118960345)

[1.7 Используемые инструменты 18](#_Toc118960346)

[1.8 Описание пользовательского интерфейса 18](#_Toc118960347)

[1.9 Приложение 19](#_Toc118960348)

# 1. **Задание №3 Оконное приложение «Отделы»**

## **Описание задачи**

Используя методы LINQ и использовать не менее двух коллекций описать базовый класс с указанными полями и методами:

* Конструктор;
* 2 дополнительных поля;
* Название отдела;
* Базовый оклад;
* Коэффициент;
* Количество сотрудников в отделе;
* ФИО начальника отдела;
* Вычисление Q = Базовый оклад \* (1 + Коэффициент / 100);
* Метод с выводом информации об объекте;

Построить класс потомок, который содержит:

* Дополнительные 2 свойства;
* Страховка сотрудника;
* Количество аварий за год
* Вычисление Qp, которая перекрывает функцию качества базового класса, выполняя вычисления по формуле если harmfulness > 0, то 2\*Q() / harmfulness (harmfulness – вредность, имеет значение от 1 до 3); иначе Qp = Q().
* Метод с выводом информации об объекте;

Логика программы с методами добавления + (перегрузка), удаления + (перегрузка), сортировка (по параметрам, ссылкам) организована в вспомогательном классе Management.

Создать проект для демонстрации работы: ввод-вывод информации об объектах базового класса и класса потомка.

Разработать модульные тесты к реализованным методам классов

## **Структура проекта**

Решение состоит из 2 проектов (Рис. 1): zd3\_EpimakhovDenis и UnitTestProdject1. Основной проект zd3\_EpimakhovDenis содержит форму, базовый класс ParentClass, класс-потомок ChildClass, а также системные файлы. Проект UnitTestProdject1 представляет из себя проект тестов, содержащую 16 модульных тестов в классе Management.

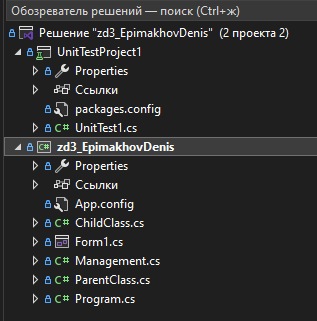


Рис. 1. Обозреватель решений

## **1.3 Описание разработанных функций**

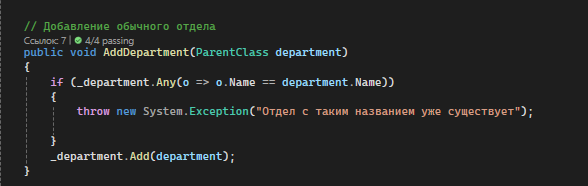
Класс (Management) с логикой состоит из 13 методов

Рис. 2. Метод добавления «обычного» отдела по ссылке.

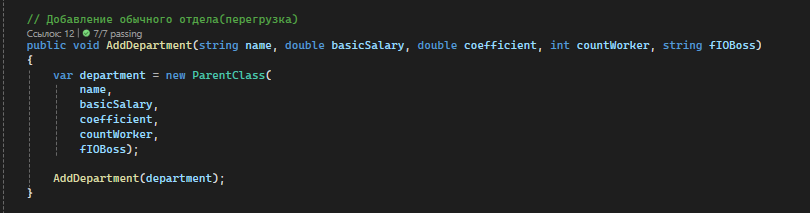


Рис. 3. Метод добавления «обычного» отдела по параметрам (перегрузка).

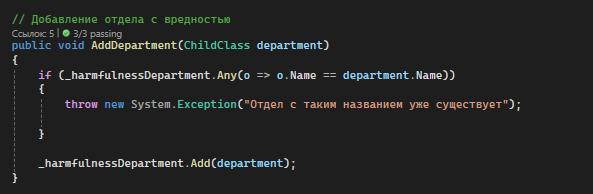


Рис. 4. Метод добавления «вредного» отдела по ссылке.

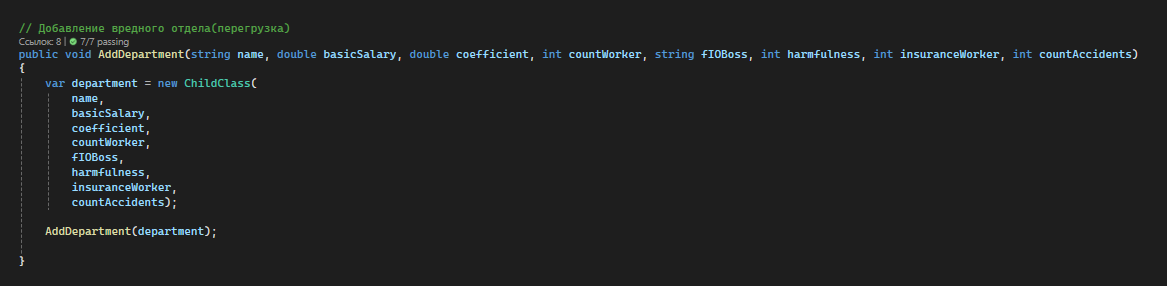


Рис. 5. Метод добавления «вредного» отдела по параметрам (перегрузка).

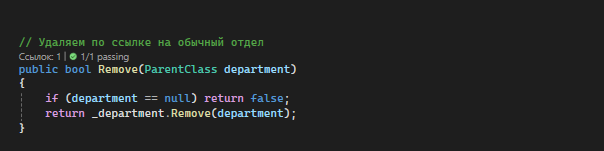
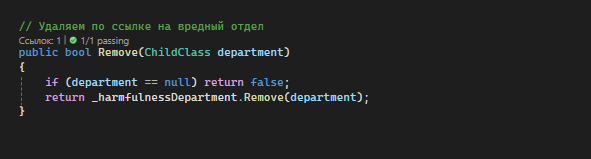


Рис. 6. Метод удаления «обычного» отдела по ссылке.

Рис. 7. Метод удаления «вредного» отдела по ссылке.

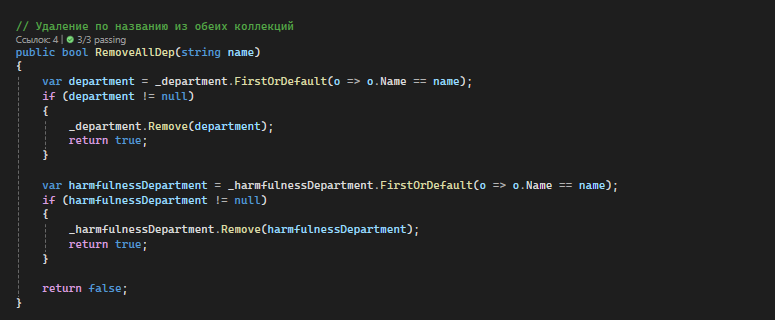


Рис. 8. Метод удаления «обычного» и «вредного» отделов по названию.

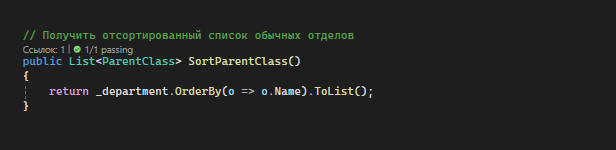


Рис. 9. Метод сортировки «обычного» отдела по названию.

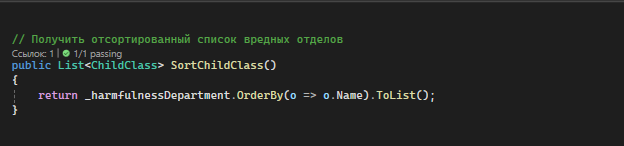
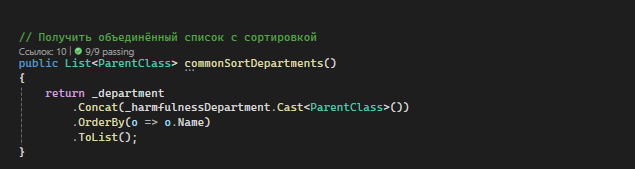


Рис. 10. Метод сортировки «вредного» отдела по названию.

Рис. 11. Метод сортировки «обычного» и «вредного» отделов по названию.

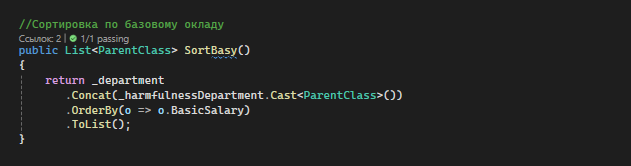


Рис. 12. Метод сортировки «обычного» и «вредного» отделов по базовому окладу.

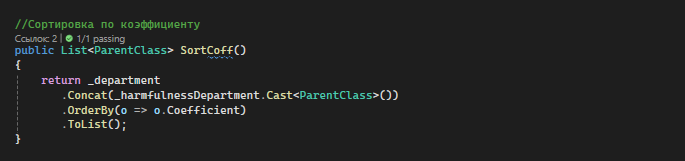


Рис. 13. Метод сортировки «обычного» и «вредного» отделов по коэффициенту.

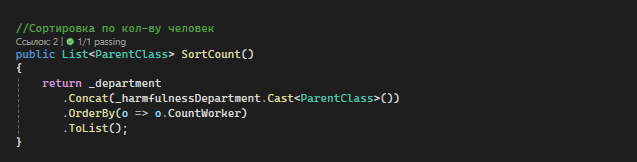


Рис. 14. Метод сортировки «обычного» и «вредного» отделов по количеству сотрудников.

## **1.4 Алгоритм решения**

Рис. 15.

## **1.5 Используемые библиотеки**

* System;
* System.Collections.Generic;
* System.ComponentModel;
* System.Data;
* System.Drawing;
* System.Linq;
* System.Text;
* System.Threading.Tasks;
* System.Windows.Forms;
* System.Windows.Forms.VisualStyles.VisualStyleElement.StartPanel;

## **1.6 Тестовые случаи**

Проект UnitTestProdject1 состоит из 16 тестов. Тесты нужны для проверки правильной работоспособности методов.

Название теста: TestMethod1;

Назначение теста: проверка добавления «обычного» отдела через AddDepartment(ParentClass);

Входные данные: объект класса ParentClass, Management;

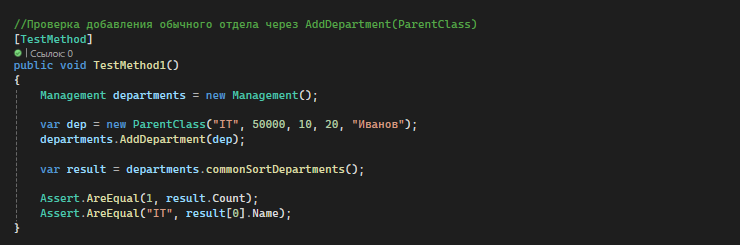
Ожидаемый результат: добавление в список departments;

Рис. 16.

Название теста: TestMethod2;

Назначение теста: проверка добавления «вредного» отдела через AddDepartment(ChildClass);

Входные данные: объект класса ChildClass, Management;

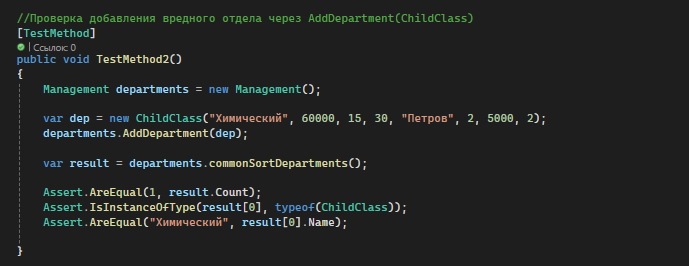
Ожидаемый результат: добавление в список departments;

Рис. 17.

Название теста: TestMethod3;

Назначение теста: проверка добавление через перегруженный метод AddDepartment() для «вредного» отдела (по параметрам);

Входные данные: объект класса Management, параметры класса ChildClass;

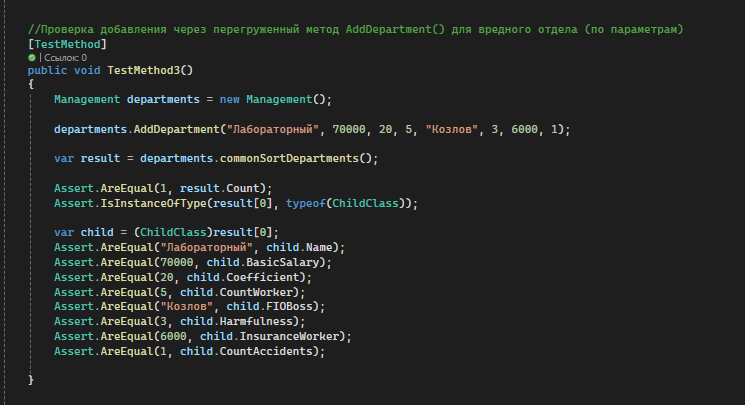
Ожидаемый результат: добавление в список departments;

Рис. 18.

Название теста: TestMethod12;

Назначение теста: проверка добавление через перегруженный метод AddDepartment() для «обычного» отдела (по параметрам);

Входные данные: объект класса Management, параметры класса ParentClass;

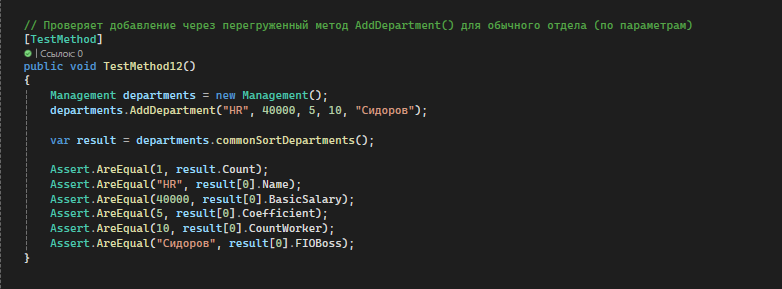
Ожидаемый результат: добавление в список departments;

Рис. 19.

Название теста: TestMethod4;

Назначение теста: проверка на запрет дубликата (два отдела с одинаковым названием);

Входные данные: объект класса ParentClass, Management;

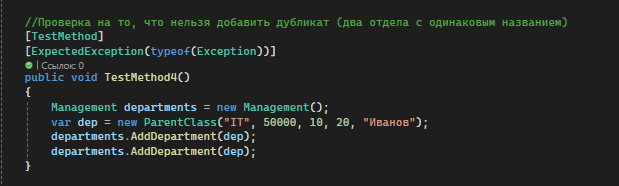
Ожидаемый результат: генерация ошибки о дублировании объектов;

Рис. 20.

Название теста: TestMethod5;

Назначение теста: проверка на удаление «обычного» отдела по названию;

Входные данные: объект класса ParentClass, Management;

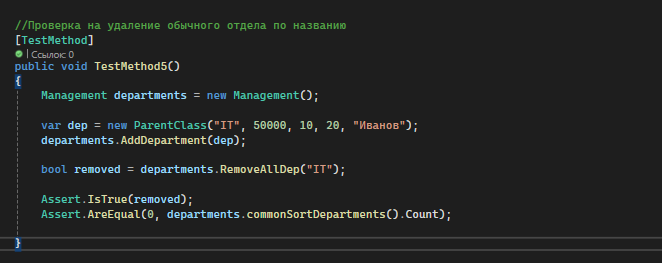
Ожидаемый результат: удаление объекта departments;

Рис. 21.

Название теста: TestMethod6;

Назначение теста: проверка на удаление «вредного» отдела по названию;

Входные данные: объект класса ChildClass, Management;

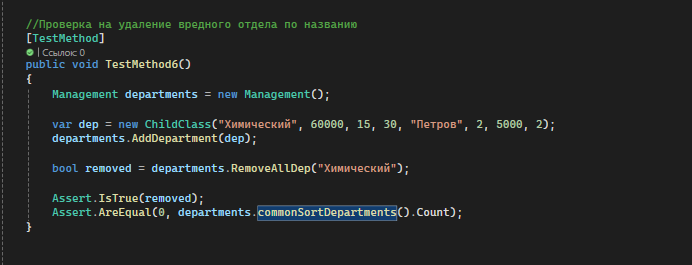
 Ожидаемый результат: удаление объекта departments;

Рис. 22.

Название теста: TestMethod7;

Назначение теста: проверка на то, что попытка удаления несуществующего отдела возвращает false;

Входные данные: объект класса Management;

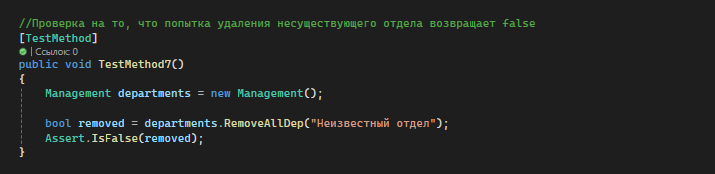
 Ожидаемый результат: возвращение false;

Рис. 23.

Название теста: TestMethod15;

Назначение теста: проверка на то, что метод Remove(ParentClass) удаляет «обычный» отдел по ссылке;

Входные данные: объект класса ParentClass, Management;

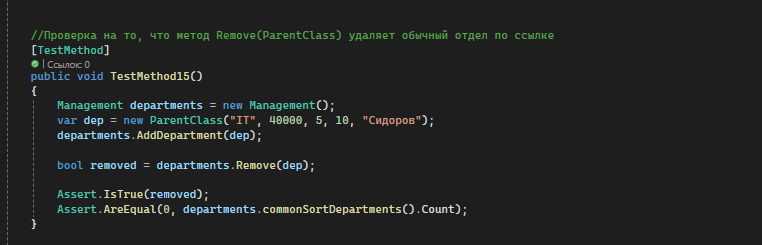
 Ожидаемый результат: удаление и возвращение true;

Рис. 24.

Название теста: TestMethod16;

Назначение теста: проверка на то, что метод Remove(ChildClass) удаляет «вредный» отдел по ссылке;

Входные данные: объект класса ChildClass, Management;

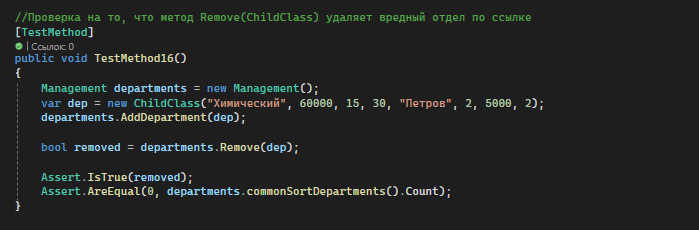
 Ожидаемый результат: удаление и возвращение true;

Рис. 25.

Название теста: TestMethod8;

Назначение теста: проверка на объединённую сортировку по названию - метод commonSortDepartments();

Входные данные: объект класса Management, параметры класса ChildClass и ParentClass;

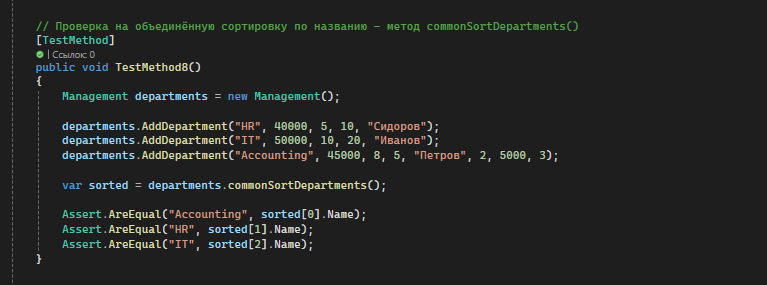
 Ожидаемый результат: сортировка объектов по названию;

Рис. 26.

Название теста: TestMethod9;

Назначение теста: проверка на объединённую сортировку по базовому окладу - метод SortBasy();

Входные данные: объект класса Management, параметры класса ChildClass и ParentClass;

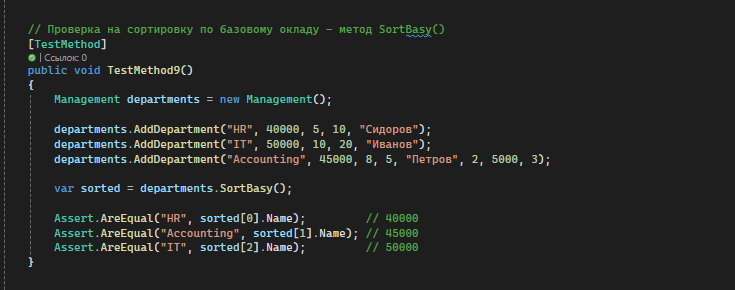
 Ожидаемый результат: сортировка объектов по базовому окладу;

Рис. 27.

Название теста: TestMethod10;

Назначение теста: проверка на объединённую сортировку по коэффициенту - метод SortCoff()

Входные данные: объект класса Management, параметры класса ChildClass и ParentClass;

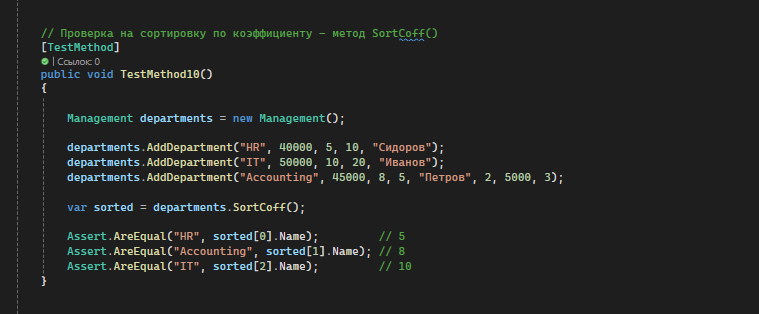
 Ожидаемый результат: сортировка объектов по коэффициенту;

Рис. 28.

Название теста: TestMethod11;

Назначение теста: проверка на объединённую по количеству сотрудников - метод SortCount();

Входные данные: объект класса Management, параметры класса ChildClass и ParentClass;

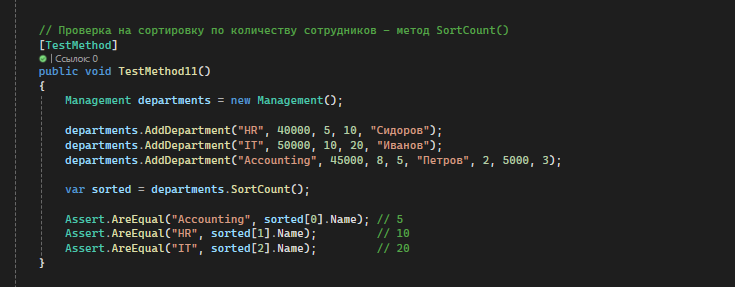
 Ожидаемый результат: сортировка объектов по количеству сотрудников;

Рис. 29.

Название теста: TestMethod13;

Назначение теста: проверка на сортировку метода - SortParentClass(). Возвращает список «обычных» отделов, отсортированный по названию;

Входные данные: объект класса Management, параметры класса ChildClass и ParentClass;

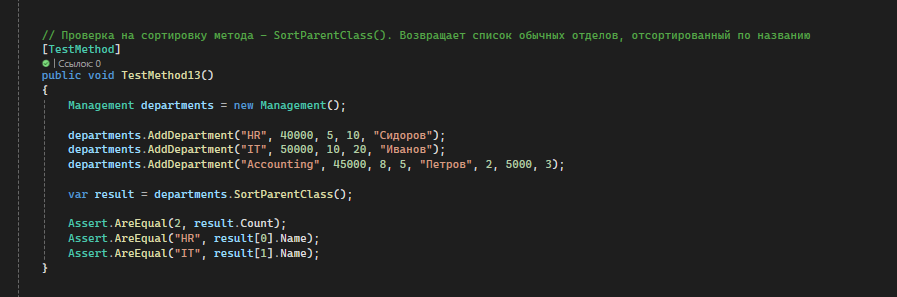
 Ожидаемый результат: возвращение списка «обычных» отделов, отсортированный по названию;

Рис. 30.

Название теста: TestMethod14;

Назначение теста: проверка на сортировку метода - SortParentClass(). Возвращает список «вредных» отделов, отсортированный по названию;

Входные данные: объект класса Management, параметры класса ChildClass и ParentClass;

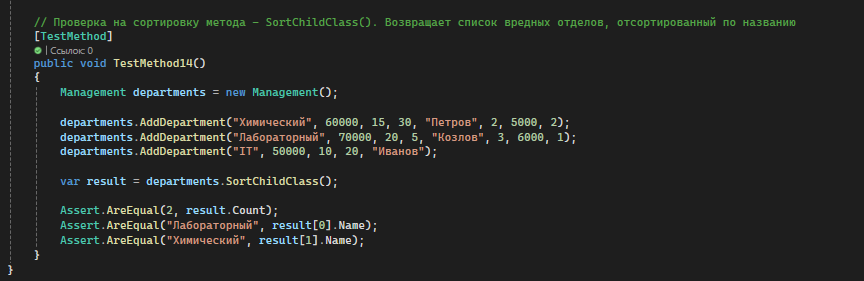
 Ожидаемый результат: возвращение списка «вредных» отделов, отсортированный по названию;

Рис. 31

Название теста: TestMethod17;

Назначение теста: проверка вычисления Q();

Входные данные: объект класса Management, ParentClass;

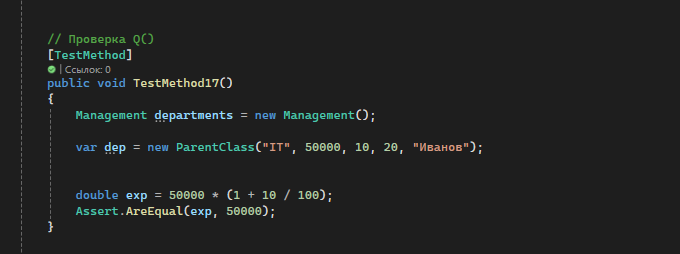
 Ожидаемый результат: возвращение Q «обычного» отдела;

Рис. 32

Название теста: TestMethod18;

Назначение теста: проверка вычисления Qp();

Входные данные: объект класса Management, ChildClass;

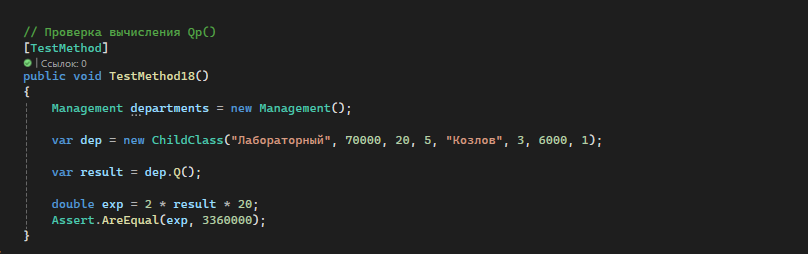
 Ожидаемый результат: возвращение Qp «вредного» отдела;

Рис. 33

## **1.7 Используемые инструменты**

Используемый язык программирования для работы: C#;

среда разработки: приложение Windows Forms (.NET Framework);

фреймворк: (.NET Framework);

## **1.8 Описание пользовательского интерфейса**

Form1.cs [Конструктор] состоит из:

* текстовое поле (label) – «Название отдела»;
* поле для ввода данных (textbox) – название отдела;
* текстовое поле (label) – «Базовый оклад»;
* поле для ввода данных (textbox) – базовый оклад;
* текстовое поле (label) – «Коэффициент»;
* поле для ввода данных (textbox) – коэффициент;
* текстовое поле (label) – «Количество сотрудников»;
* поле для ввода данных (textbox) – количество сотрудников;
* текстовое поле (label) – «ФИО начальника»;
* поле для ввода данных (textbox) – ФИО начальника;
* поле для обозначения наличия (checkBox) – Вредность;
* текстовое поле (label) – «Уровень вредности»;
* целочисленное поле (numericUpDown) – уровень вредности;
* текстовое поле (label) – «Количество аварий за год»;
* целочисленное поле (numericUpDown) – количество аварий за год;
* текстовое поле (label) – «Страховка сотрудника»;
* целочисленное поле (numericUpDown) – страховка сотрудника;
* табличное поле (DataGridView) – вывод информации об отделе;
* меню (menuStrip) – пункты меню (Добавить отдел; Удалить отдел; Сортировка по базовому окладу; Сортировка по коэффициенту; Сортировка по количеству людей);

## **1.9 Приложение**

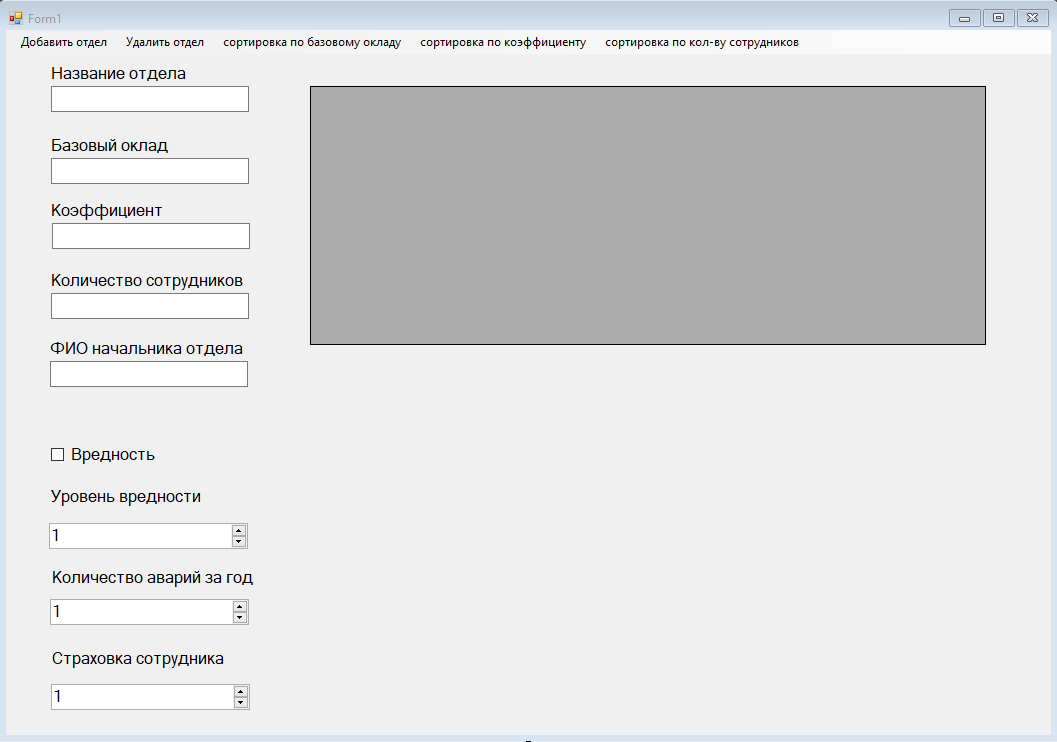
****Form1.cs [Конструктор]

Рис. 34.

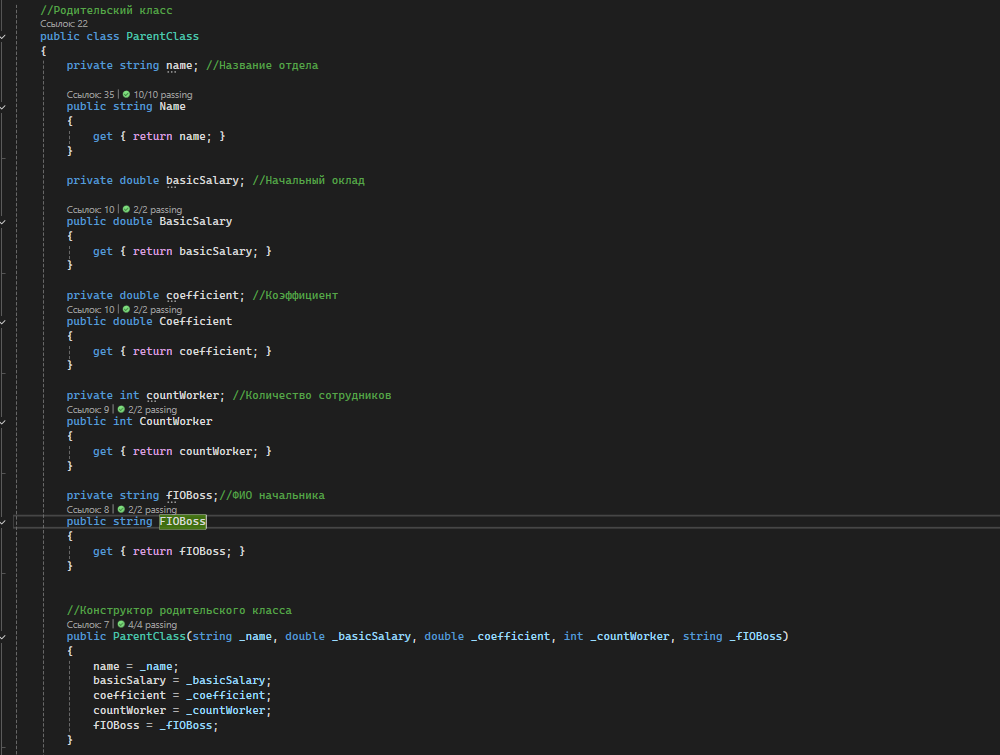
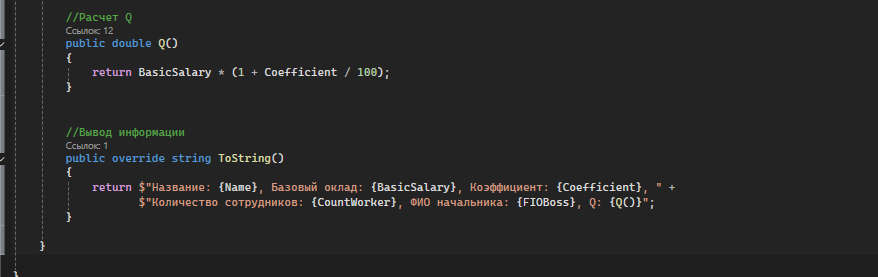
ParentClass.cs - родительский класс

Рис. 35.

Рис. 36.

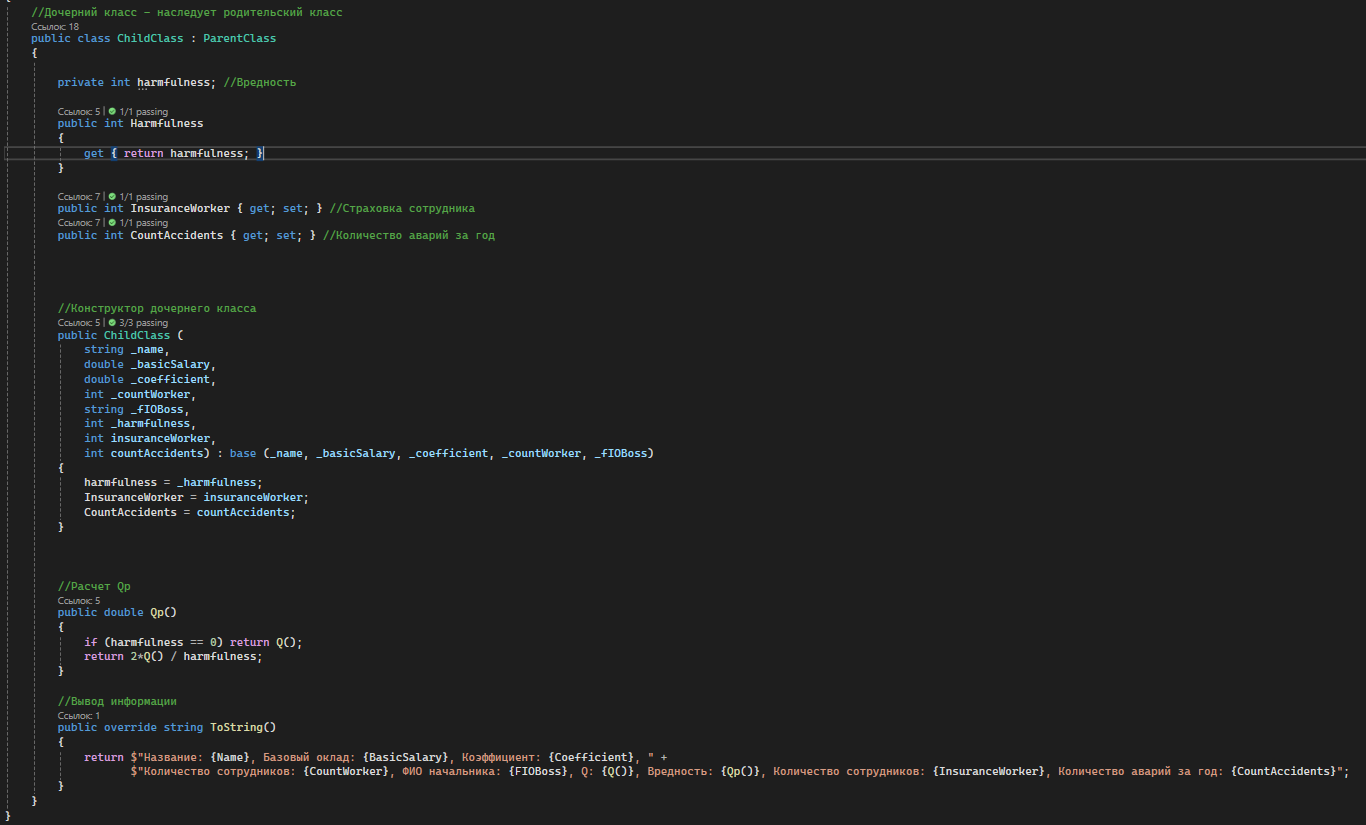
 ChildClass.cs – дочерний класс

Рис. 37.

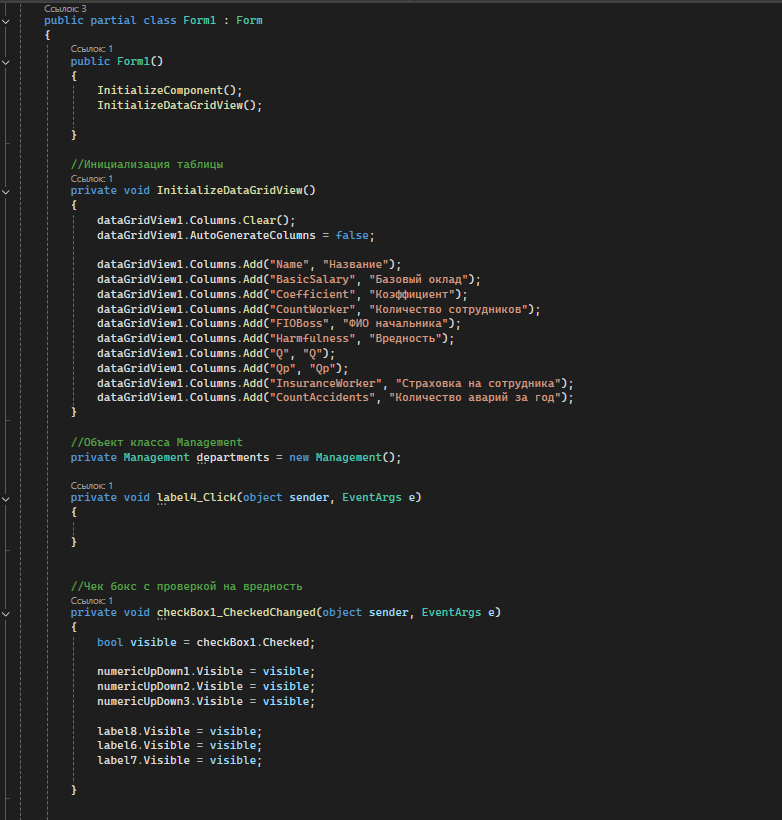
Form1.cs

Рис. 38.

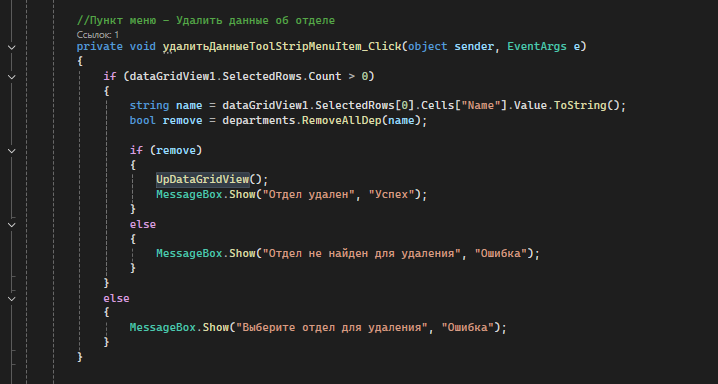


Рис. 39.

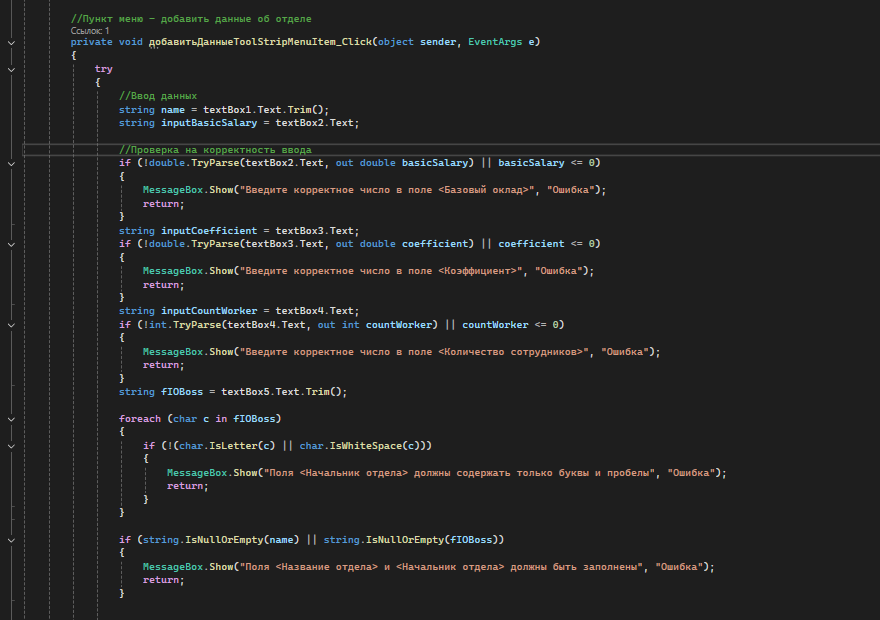
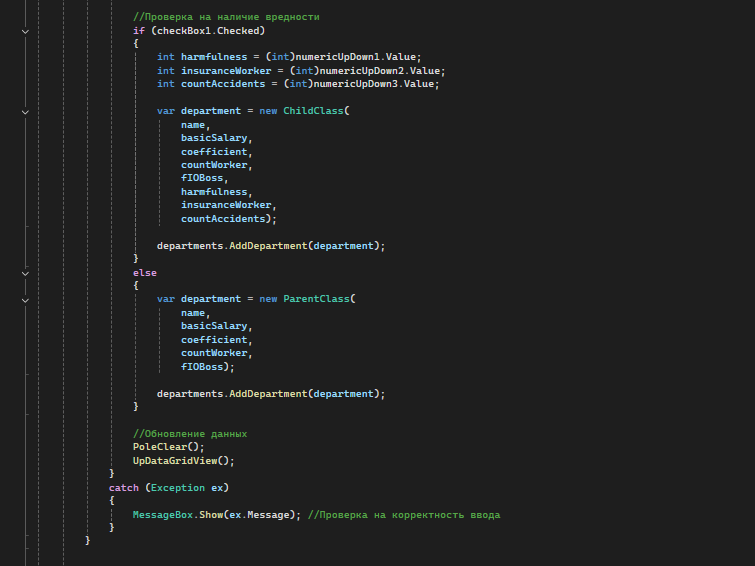


Рис. 41.

Рис. 40.

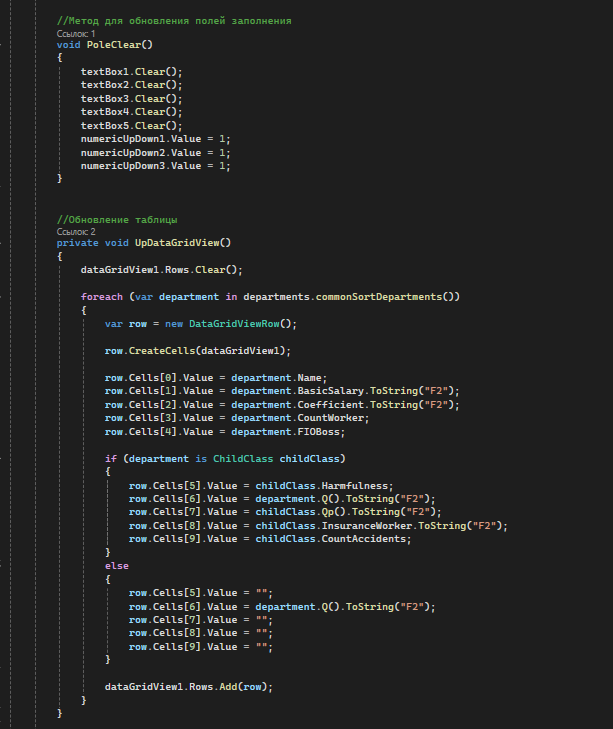
Рис. 42.

Рис. 43.

Рис. 44.

Рис. 45.

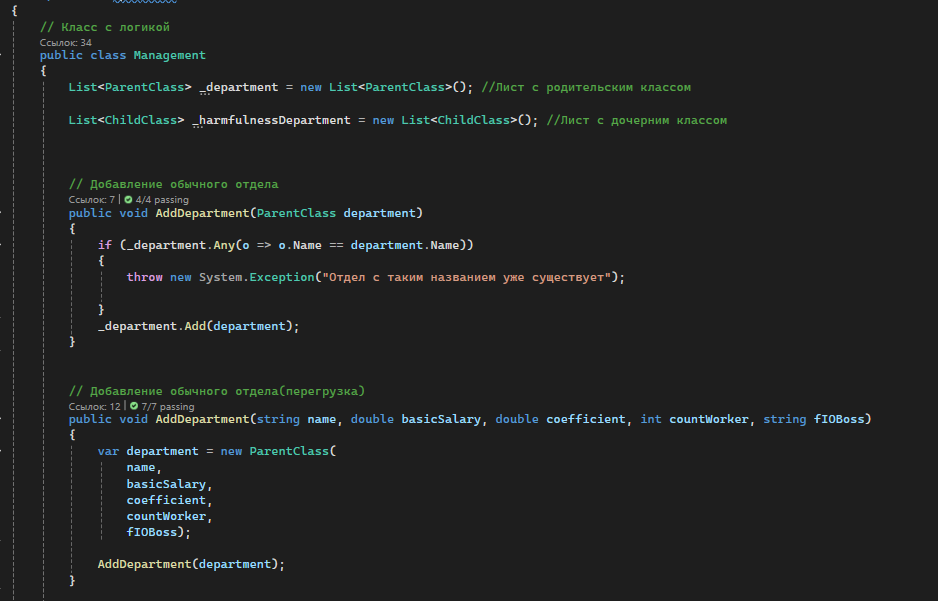
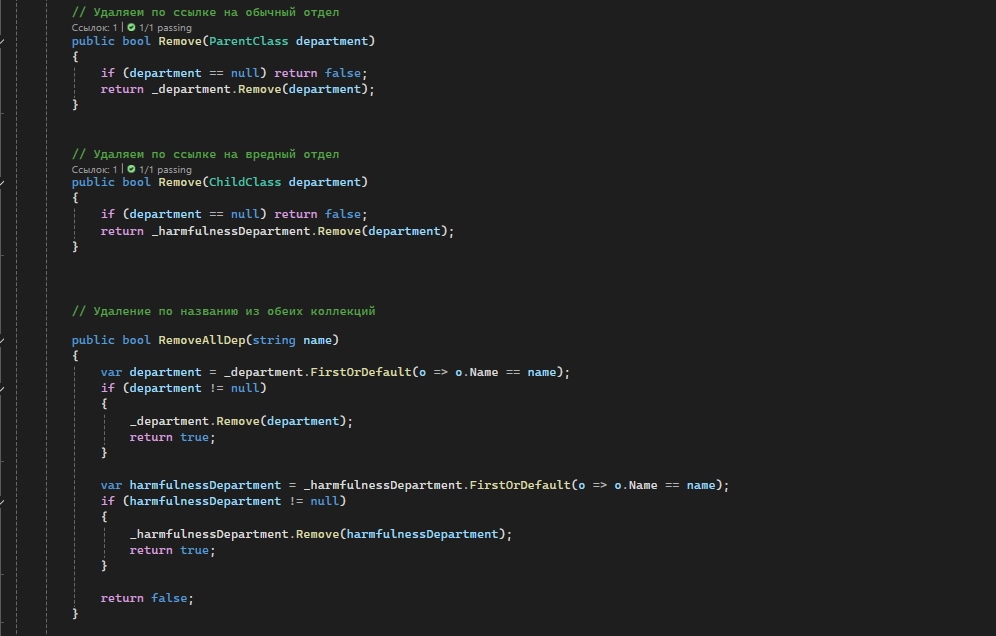
Класс Management.cs

Рис.46.

Рис.47.

Рис. 48.

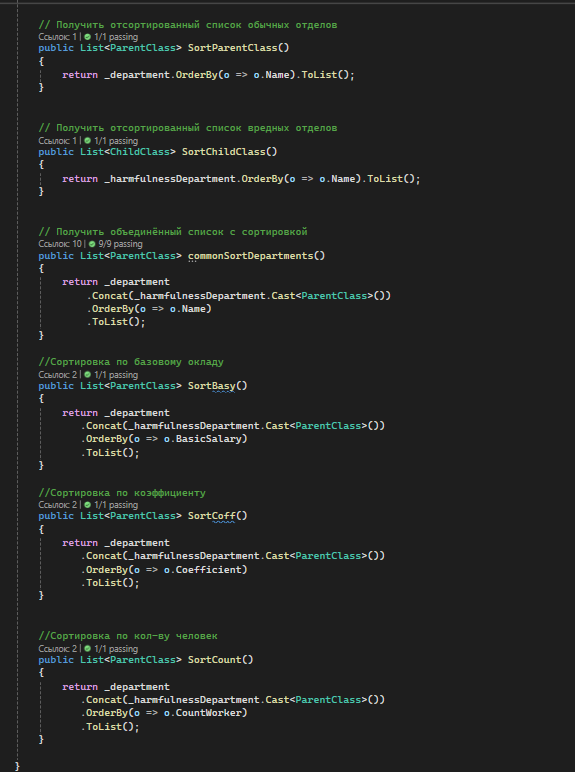


Рис. 49.