

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.02

Выполнил: Коньков Арсений Андреевич

Группа: ПР-22

Преподаватель: Мирошниченко Г.В.

2025

Содержание

# [1.](#_Toc118960339) **[Задание №3 Индивидуальное задание «Кабель»..............................](#_Toc118960339)**[3](#_Toc118960339)

[1.1 Описание задачи 3](#_Toc118960340)

[1.2 Структура проекта 3](#_Toc118960341)

[1.3 Описание разработанных функций 3](#_Toc118960342),4

[1.4 Алгоритм решения 5](#_Toc118960343)

[1.5 Используемые библиотеки 6](#_Toc118960344)

[1.6 Тестовые случаи 6](#_Toc118960345)

[1.7 Используемые инструменты 6](#_Toc118960346)

[1.8 Описание пользовательского интерфейса 6](#_Toc118960347)

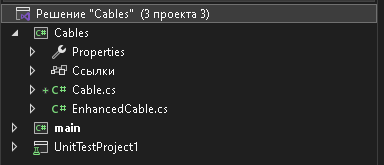
[1.9 Приложение (pr screen экранов) 7](#_Toc118960348)

# 1. **Задание №3 Индивидуальное задание «Кабель»**

## Описание задачи

Используя методы Linq и коллекции, описать базовый класс и класс потомка (кабель, качества), создать проект для демонстрации (WinForms) и написать модульное тестирование

## Структура проекта



Проект Cables - Библиотека классов(.NET Framework), в проект входит базовый класс cable.cs и класс наследник EnhancedCable.

Проект main - Windows Forms (.NET Framework), в проект входит приложение для взаимодействия с классами.

Проект UnitTestProject1 - Проект модульного тестирования(.NET Framework), в проект входят модульные тесты.

## 1.3 Описание разработанных функций

(Привести методы. Описать их назначение, входные и выходные значения)

CalculateQuality() - расчитывает качество, входные данные диаметр, количество жил кабеля. (базовый класс)

CalculateQuality() - Расчитывает улучшенное качество, входные данные результат функции CalculateQuality() из базового класса, выходные данные если у кабеля есть оплетка, тогда CalculateQuality() \* 2, иначе CalculateQuality() \* 0,7.

DisplayInfo() - Показывает информацию о кабеле

AddCable() - Метод для добавление в список обьект кабель, входные данные коллекция List и обьект кабель

AddCable() - Перегрузка метода для добавления в список обьект кабель, входные данные коллекция List и строка “Тип кабеля”, целое число “Количество жил”, число с плавающей запятой “Диаметр”, строка “Производитель”, число с плавающей запятой “Цена за метр”

RemoveCable() - Метод для удаления обьекта кабель из списка, входные данные коллекция List и обьект кабель

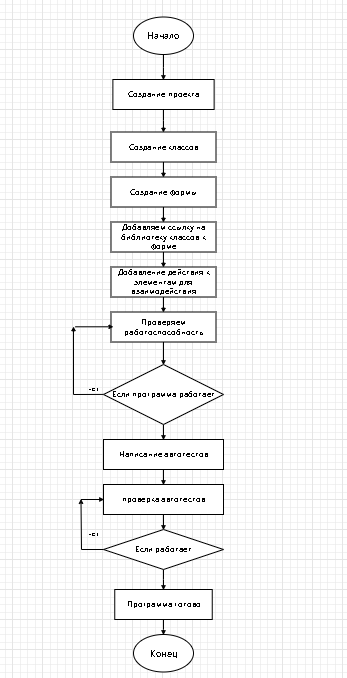
RemoveCable() - Перегрузка метода для удаления обьекта кабель из списка, входные данные коллекция List и строка “Тип кабеля”, целое число “Количество жил”, число с плавающей запятой “Диаметр”, строка “Производитель”, число с плавающей запятой “Цена за метр”

AddEnhancedCable() - Метод для добавления обьект кабель с качествами, входные данные коллекция List и строка “Тип кабеля”, целое число “Количество жил”, число с плавающей запятой “Диаметр”, строка “Производитель”, число с плавающей запятой “Цена за метр”, логическое значение “Наличие оплетки”, строка “Цвет”.

RemoveEnhancedCable() - Метод для удаления обьекта кабель с качествами из списка, входные данные коллекция List и строка “Тип кабеля”, целое число “Количество жил”, число с плавающей запятой “Диаметр”, строка “Производитель”, число с плавающей запятой “Цена за метр”, логическое значение “Наличие оплетки”, строка “Цвет”.

## 1.4 Алгоритм решения

(Описать способ решения задачи с помощью блок-схемы)



## 1.5 Используемые библиотеки

System.Collections.Generic - для работы с коллекциями.

System.Linq - Язык запросов.

## 1.6 Тестовые случаи

CableQualityCalculation\_ShouldReturnCorrectValue - Тест на правильность

расчёта, выходные данные диаметр = 8, количество жил = 4, ожидаемые

результат 2.0.

EnhancedCableQuality\_WithBraiding\_ShouldDoubleQuality() - Тест на

правильность расчёта, входные данные диаметр = 8, количество жил = 4,

ожидаемый результат 4.0.

## 1.7 Используемые инструменты

Для данной работы использовался язык программирования C#, в среде разработки Visual Studio, использован фреймворк .NET Framework.

## 1.8 Описание пользовательского интерфейса

(Основные пункты меню на экране с кратким описанием того, как это должно работать)

Для пользовательского интерфейса использовались элементы - ListBox, TextBox, ComboBox, NumericUpDown, Label, ContextMenuStrip

Для создания объекта кабель нужно заполнить данные поля соответствующими значениями, затем нажать правой кнопкой мыши, в любом месте, появится меню, затем нажать на кнопку “Добавить”. После этого объект будет создан, в ListBox появится значение подписанный как тип объекта.

Для удаления объекта кабель нужно выбрать кабель из списка, нажать правой кнопкой мыши в любом место, появится меню, затем нажать кнопку “Удалить”. После этого объект будет удалён.

Для выхода из приложения нужно нажать правой кнопкой мыши в любом месте, появится меню, затем нажать кнопку “Выход”, после этого процесс программы будет завешен.

1.9 Приложение

