

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.02

Выполнил: Курдин Иван

Группа: ПР-22

Преподаватель: Мирошниченко Г.В.

2025

Содержание

[1.](#_Toc118960339) **[Задание №1 Мобильное приложение «Дневник тренировок»](#_Toc118960339)** [3](#_Toc118960339)

[1.1 Описание задачи 3](#_Toc118960340)

[1.2 Структура проекта 3](#_Toc118960341)

[1.3 Описание разработанных функций 3](#_Toc118960342)

[1.4 Алгоритм решения 3](#_Toc118960343)

[1.5 Используемые библиотеки 3](#_Toc118960344)

[1.6 Тестовые случаи 3](#_Toc118960345)

[1.7 Используемые инструменты 3](#_Toc118960346)

[1.8 Описание пользовательского интерфейса 3](#_Toc118960347)

[1.9 Приложение (pr screen экранов) 3](#_Toc118960348)

1. **Задание №3 Индивидуальное задание**

## **Описание задачи**

Поля и функция «качества» Q базового Класса:

Компьютер:

- наименование процессора;

-тактовая частота процессора (МГц);

- объем оперативной памяти

(Мб). Q = (0,3·частота) + Память.

Поля и функция «качества» Qp класса потомка:

P: объём винчестера (Гб)

Qp=Q+0,5·Р.

Используя методы Linq и коллекции (использовать не менее 2-х видов коллекций) Описать базовый класс с указанными в задании полями и методами:

● Конструктор;

● Добавить 2 своих поля;

● Функция, которая определяет качество объекта – Q по заданной формуле;

● Вывод информации об объекте.

● Реализуйте и проверьте методы для добавления и удаления

объекта базового класса (сделать 2 перегрузки!);

Построить класс потомок, который содержит:

● Дополнительное поле Р;

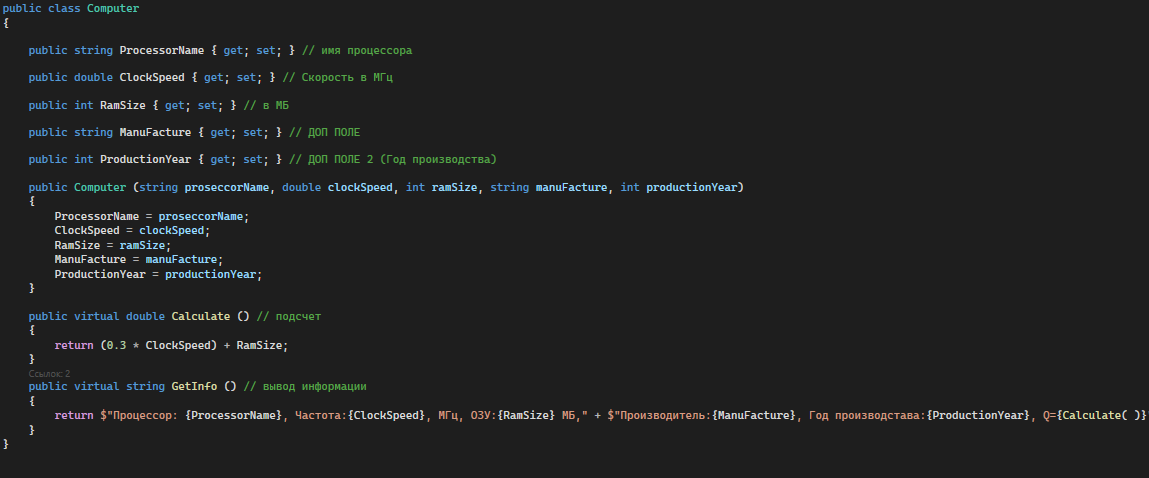
● Добавить еще 1-2 свойства

● Функция, которая определяет «качество» объекта класса потомка – Qp, которая перекрывает функцию качества базового класса, выполняя вычисления по новой формуле.

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

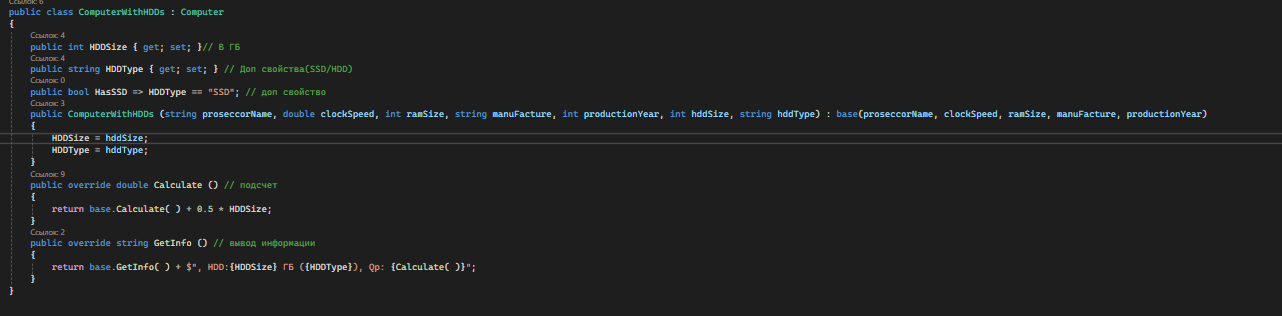
Разработать модульные тесты к реализованным методам классов

## **Структура проекта**



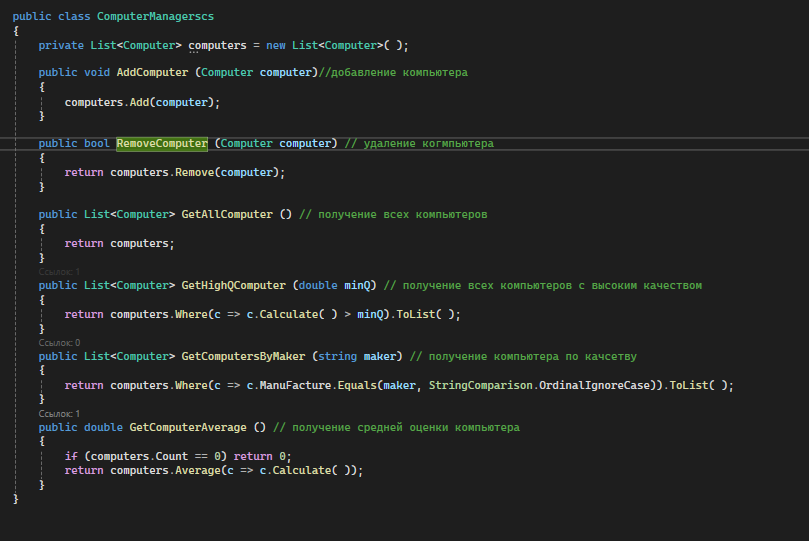
Класс Computer

* + Назначение: Базовый класс, представляющий компьютер.
  + Свойства:
    - ProcessorName — название процессора.
    - ClockSpeed — тактовая частота.
    - RansSize — размер оперативной памяти.
    - Manufacture — производитель.
    - ProductionYear — год производства.
  + Методы:
    - Calculate() — вычисляет показатель качества компьютера.
    - GetInfo() — возвращает строку с информацией о компьютере.



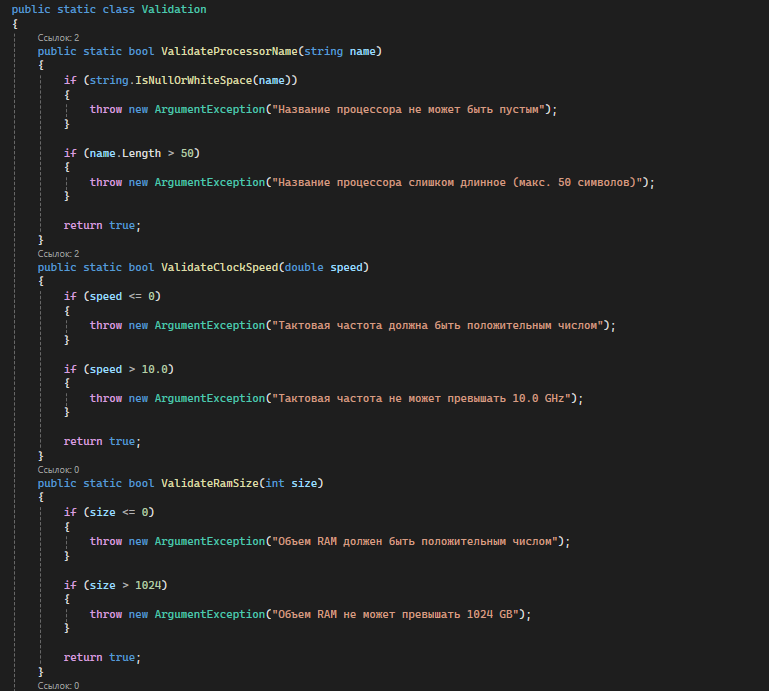
Класс ComputerWithHDDs (наследуется от Computer)

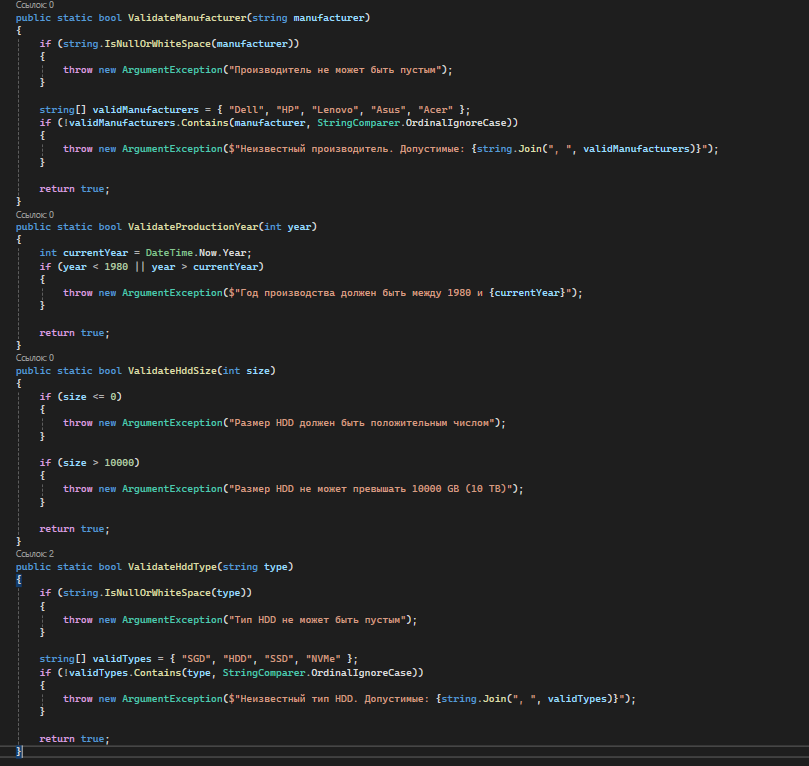
* + Назначение: Расширяет функциональность базового класса, добавляя информацию о жестком диске.
  + Дополнительные свойства:
    - HDDSize — размер жесткого диска.
    - HDDType — тип жесткого диска.
    - HasSize — проверяет, является ли тип диска "SSD".
  + Методы:
    - Переопределенные Calculate() и GetInfo() для учета характеристик жесткого диска.



Класс ComputerManagerscs

* + Назначение: Управляет коллекцией компьютеров.
  + Методы:
    - AddComputer(), RemoveComputer() — добавление и удаление компьютеров.
    - GetAllComputer() — возвращает все компьютеры.
    - GetHighQComputer() — возвращает компьютеры с качеством выше заданного.
    - GetComputersByMaker() — фильтрация по производителю.
    - GetComputerAverage() — вычисляет среднее качество компьютеров.





Класс Validation - основной класс, содержащий все методы валидации.

* + Методы:
    - ValidateProcessorName - проверяет название процессора на пустоту и длину (макс. 50 символов).
    - ValidateClockSpeed - проверяет тактовую частоту процессора (должна быть от 0 до 10 GHz).
    - ValidateRamSize - проверяет объем RAM (должен быть от 1 до 1024 GB).
    - ValidateManufacturer - проверяет производителя на соответствие списку допустимых значений (Dell, HP, Lenovo, Asus, Acer).
    - ValidateProductionYear - проверяет год производства (должен быть между 1980 и текущим годом).
    - ValidateHddSize - проверяет размер HDD (должен быть от 1 до 10000 GB).
    - ValidateHddType - проверяет тип HDD на соответствие списку допустимых значений (SGD, HDD, SSD, NVNe).

## **1.3 Описание разработанных функций**

Класс ComputerManager

AddComputer

* Назначение: Добавляет компьютер в коллекцию.
* Входные данные:
  + computer (Computer) - объект компьютера для добавления.
* Выходные данные: Нет.

RemoveComputer

* Назначение: Удаляет компьютер из коллекции.
* Входные данные:
  + computer (Computer) - объект компьютера для удаления.
* Выходные данные:
  + bool - true, если компьютер был удален, false если не найден.

GetAllComputers

* Назначение: Возвращает все компьютеры в коллекции.
* Входные данные: Нет.
* Выходные данные:
  + List<Computer> - список всех компьютеров.

GetHighQComputers

* Назначение: Возвращает компьютеры с качеством выше заданного.
* Входные данные:
  + minQ (double) - минимальное значение качества.
* Выходные данные:
  + List<Computer> - список компьютеров, где Calculate() > minQ.

GetComputersByMaker

* Назначение: Возвращает компьютеры указанного производителя.
* Входные данные:
  + maker (string) - название производителя.
* Выходные данные:
  + List<Computer> - список компьютеров данного производителя.

GetComputerAverage

Назначение: Вычисляет среднее значение качества компьютеров.

Входные данные: Нет.

Выходные данные:

double - среднее значение Calculate() для всех компьютеров, или 0 если коллекция пуста.

## **1.4 Алгоритм решения**

(Описать способ решения задачи с помощью блок-схемы)

## **1.5 Используемые библиотеки**

## **LibraryCompouter**

## **1.6 Тестовые случаи**

Класс Validation

ValidateProcessorName

* Назначение: Проверяет валидность названия процессора.
* Входные данные:
  + name (string) - название процессора для проверки.
* Выходные данные:
  + bool - true, если проверка пройдена.
  + Исключение ArgumentException, если:
    - название пустое или состоит из пробелов
    - длина названия превышает 50 символов

ValidateClockSpeed

* Назначение: Проверяет валидность тактовой частоты процессора.
* Входные данные:
  + speed (double) - тактовая частота в GHz.
* Выходные данные:
  + bool - true, если проверка пройдена.
  + Исключение ArgumentException, если:
    - частота ≤ 0
    - частота > 10.0 GHz

ValidateRamSize

* Назначение: Проверяет валидность объема оперативной памяти.
* Входные данные:
  + size (int) - объем RAM в GB.
* Выходные данные:
  + bool - true, если проверка пройдена.
  + Исключение ArgumentException, если:
    - размер ≤ 0
    - размер > 1024 GB

ValidateManufacturer

* Назначение: Проверяет валидность производителя компьютера.
* Входные данные:
  + manufacturer (string) - название производителя.
* Выходные данные:
  + bool - true, если проверка пройдена.
  + Исключение ArgumentException, если:
    - строка пустая
    - производитель не входит в список допустимых (Dell, HP, Lenovo, Asus, Acer)

ValidateProductionYear

* Назначение: Проверяет валидность года производства.
* Входные данные:
  + year (int) - год производства.
* Выходные данные:
  + bool - true, если проверка пройдена.
  + Исключение ArgumentException, если:
    - год < 1980
    - год > текущего года

ValidateHddSize

* Назначение: Проверяет валидность размера жесткого диска.
* Входные данные:
  + size (int) - размер HDD в GB.
* Выходные данные:
  + bool - true, если проверка пройдена.
  + Исключение ArgumentException, если:
    - размер ≤ 0
    - размер > 10000 GB (10 TB)

ValidateHddType

* Назначение: Проверяет валидность типа жесткого диска.
* Входные данные:
  + type (string) - тип HDD.
* Выходные данные:
  + bool - true, если проверка пройдена.
  + Исключение ArgumentException, если:
    - строка пустая
    - тип не входит в список допустимых (SGD, HDD, SSD, NVNe)

## **1.7 Используемые инструменты**

C#,Windows form, .NET Framework 4.7.2

## **1.8 Описание пользовательского интерфейса**

Label 1 - 4 = вывод статистику на экран.

TextBox = Сюда вносятся информация которая выводится в datagridview

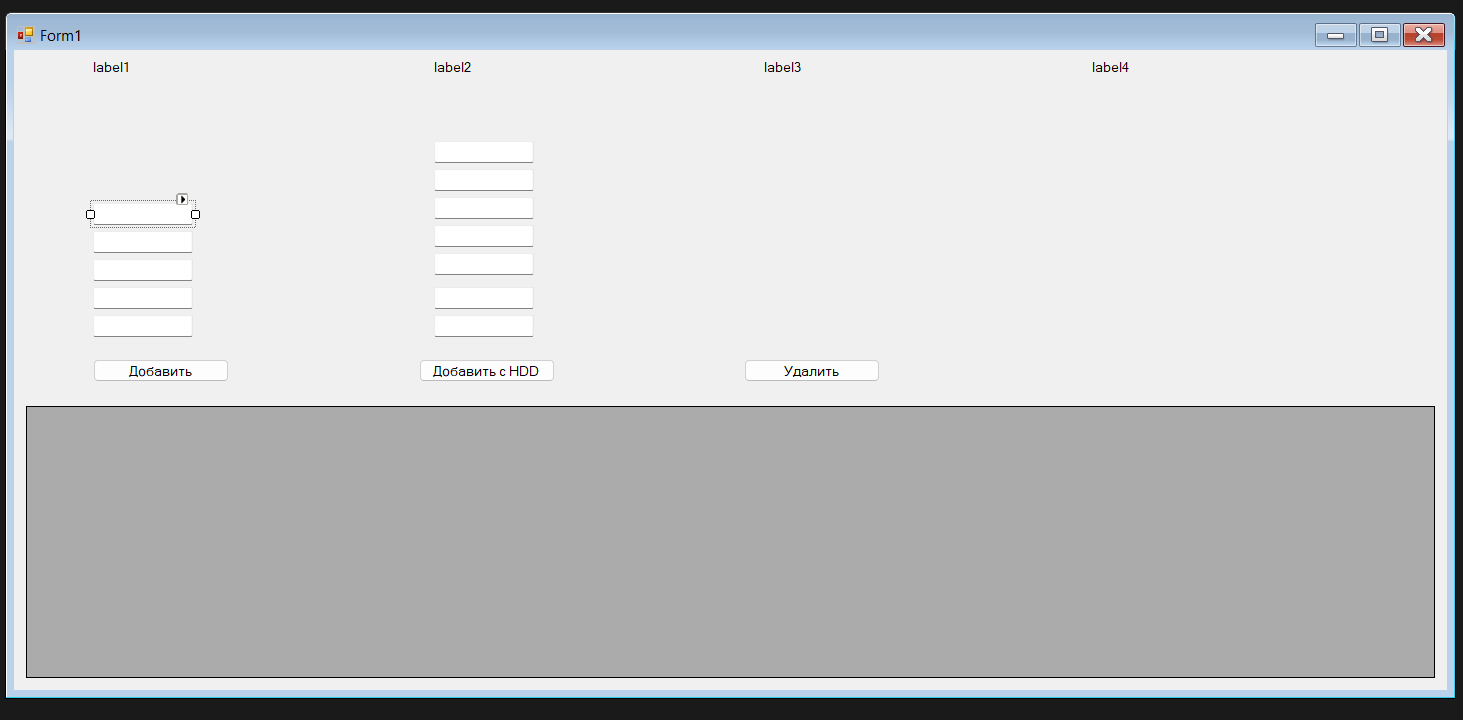
Buttonadd - кнопка добавления

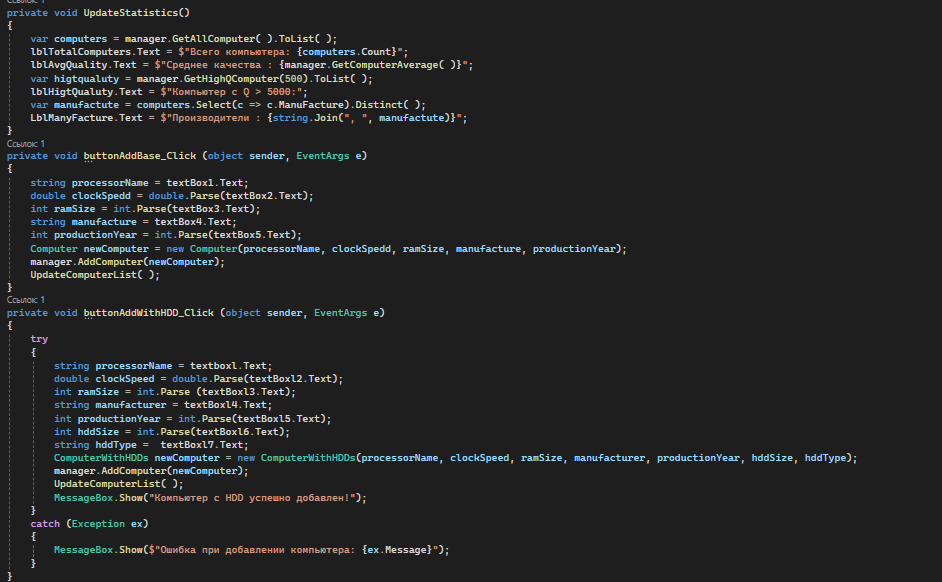
Buttonaddhdd - кнопка добавления с hdd

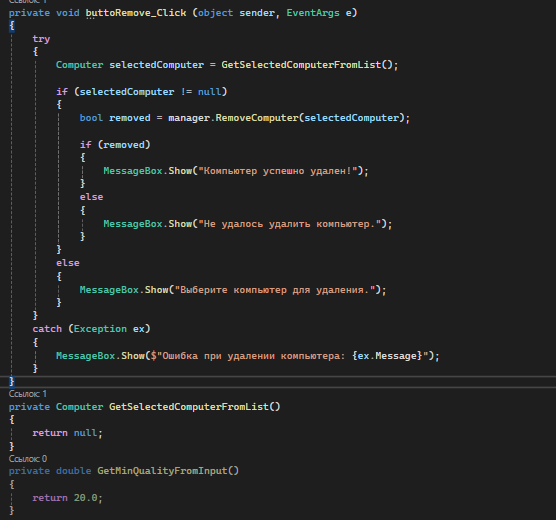
Buttondelete - удаления компьютера из datagridview

Datagrodview - список с компьютерами

## **1.9 Приложение (pr screen экранов)**

****

****

****