МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Омский государственный технический университет»
Факультет информационных технологий и компьютерных систем
Кафедра «Информатика и вычислительная техника»

ОТЧЕТ

по лабораторной работе №2
по дисциплине «Объектно-ориентированное программирование»

Выполнил студент
группы ИВТ-212

Проверил ст. преподаватель
Степанов П.П.
_____

Омск, 2022

Задание:

Используя Visual Studio, создать проект по шаблону Console Application.

Требуется:

- создать класс, представляющий учебный класс ClassRoom;
- создать класс Pupil (ученик).
- В теле класса создать методы: Study(); Read(); Write(); Relax();
- создать три производных класса на основе базового класса Pupil: ExcelentPupil; GoodPupil; BadPupil;
- – преобразовать перечисленные три производных класса запечатанными и переопределить каждый из методов в зависимости от успеваемости ученика.

Конструктор класса ClassRoom принимает аргументы типа Pupil, класс должен состоять из четырех учеников. 45 Необходимо предусмотреть возможность того, что пользователь может передать два или три аргумента.

- Реализовать статическое поле «Количество учеников» в классе ClassRoom.
- К классу Pupil реализовать виртуальное свойство, доступное только для чтения GetCurrentGrade, позволяющее возвращать текущую оценку ученика (случайное число от 2 до 5).
- В классах-наследниках перегрузить свойство GetCurrentGrade так, чтобы вероятность получения хорошей оценки у способных учеников была выше.
- В классе ClassRoom реализовать свойство GetRoundGrade, вычисляющее средний балл по группе.
- Написать 10 unit-тестов, проверяющих корректность работы программы.

Реализация класса ClassRoom:

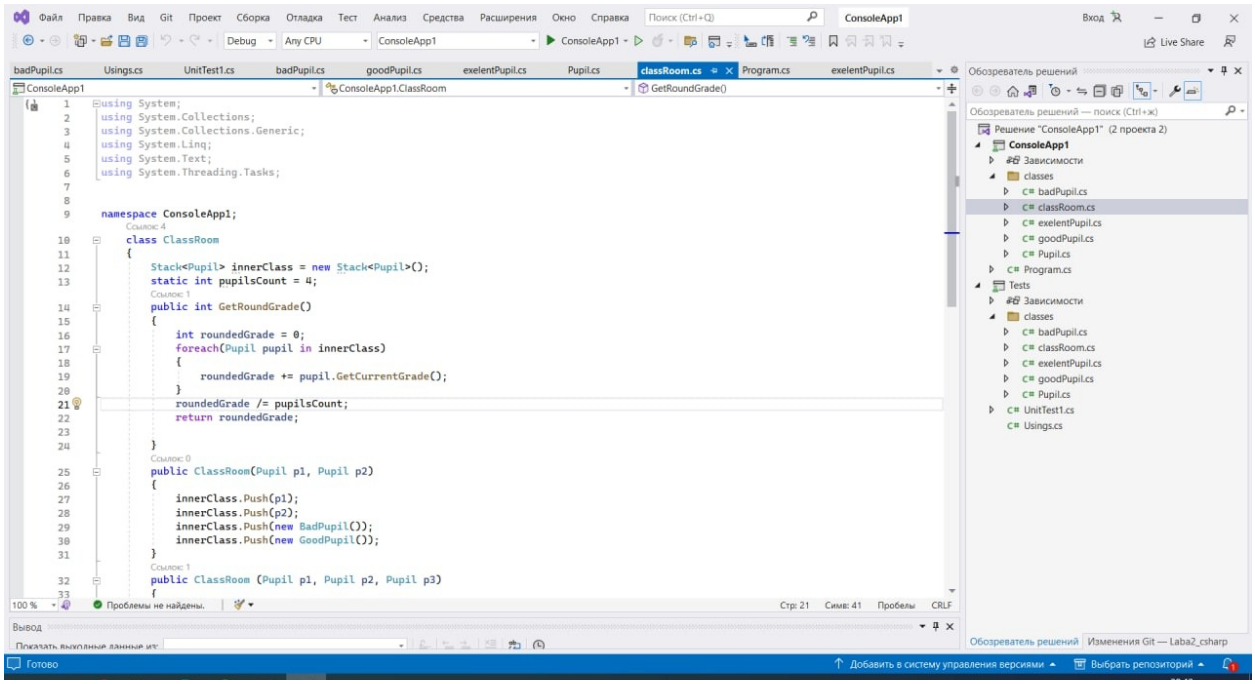


Рис. 1 – Класс ClassRoom

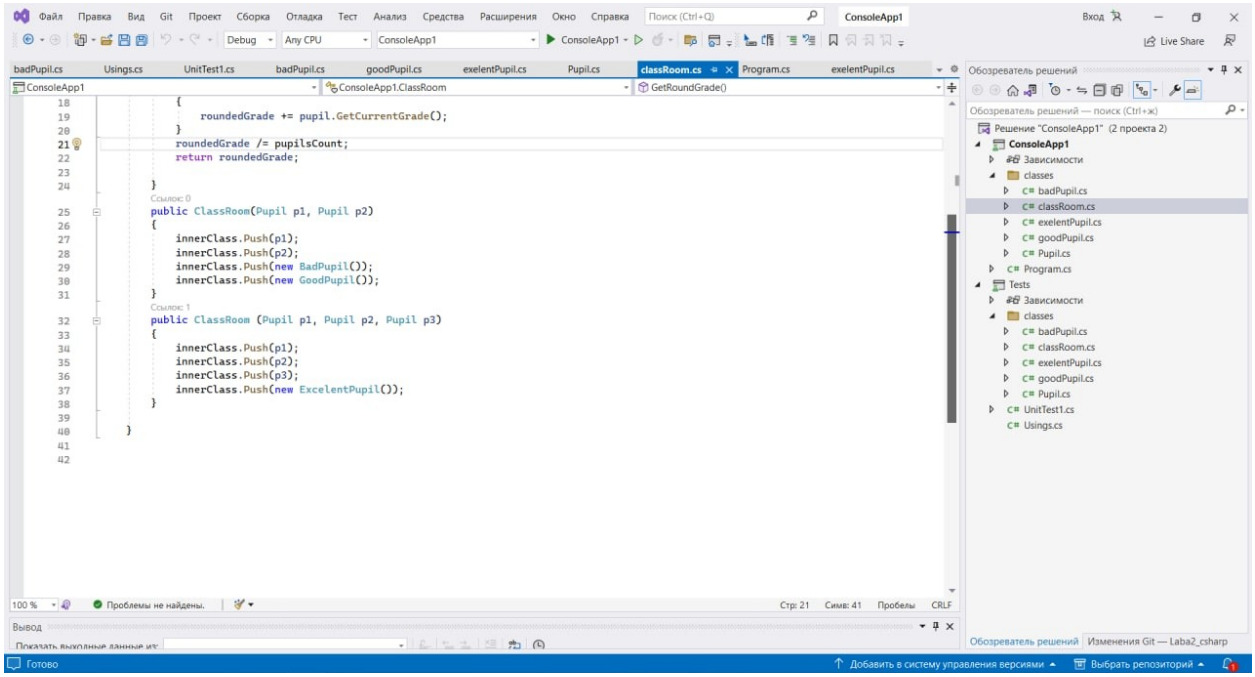
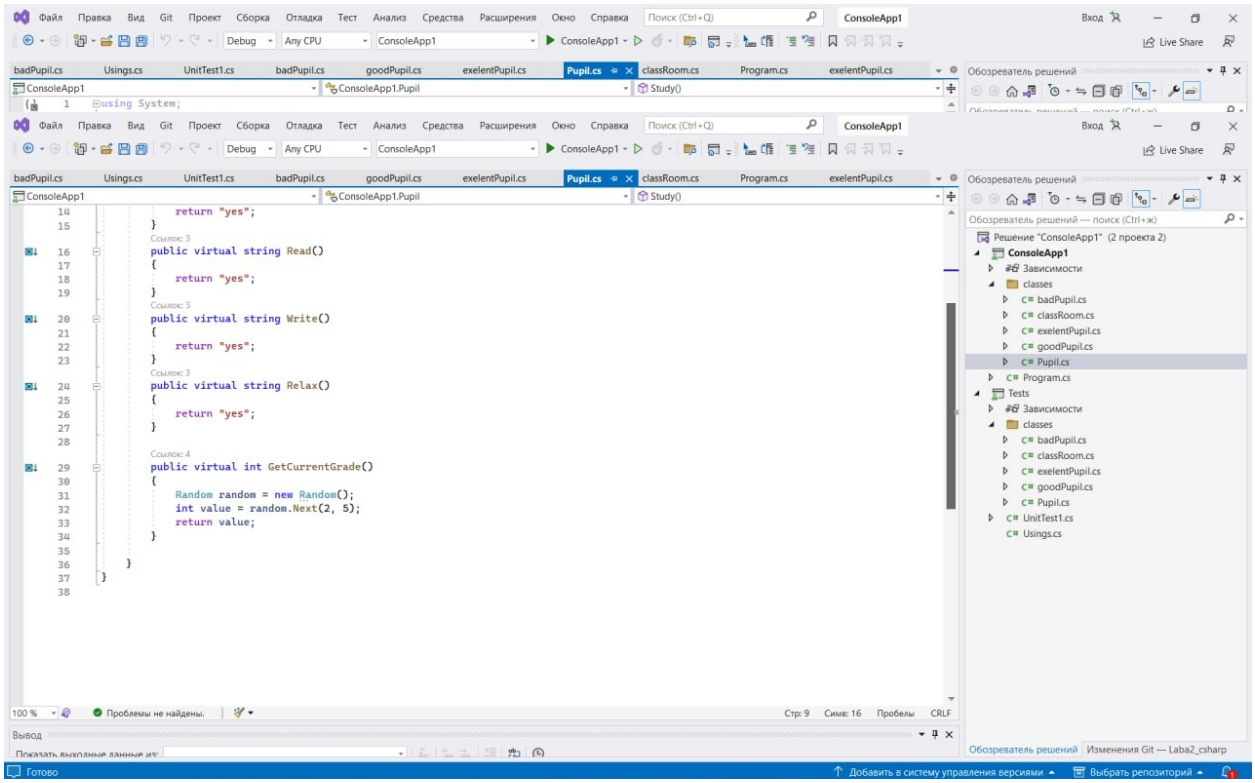


Рис. 2 - Класс ClassRoom

Реализация базового класса Pupil:



```
using System;

namespace ConsoleApp1
{
    public class Pupil
    {
        public virtual string Read()
        {
            return "yes";
        }
        public virtual string Write()
        {
            return "yes";
        }
        public virtual string Relax()
        {
            return "yes";
        }

        public virtual int GetCurrentGrade()
        {
            Random random = new Random();
            int value = random.Next(2, 5);
            return value;
        }
    }
}
```

Рис.3 – Класс Pupil

Рис. 4 – Класс Pupil

Реализация производного класса GoodPupil:

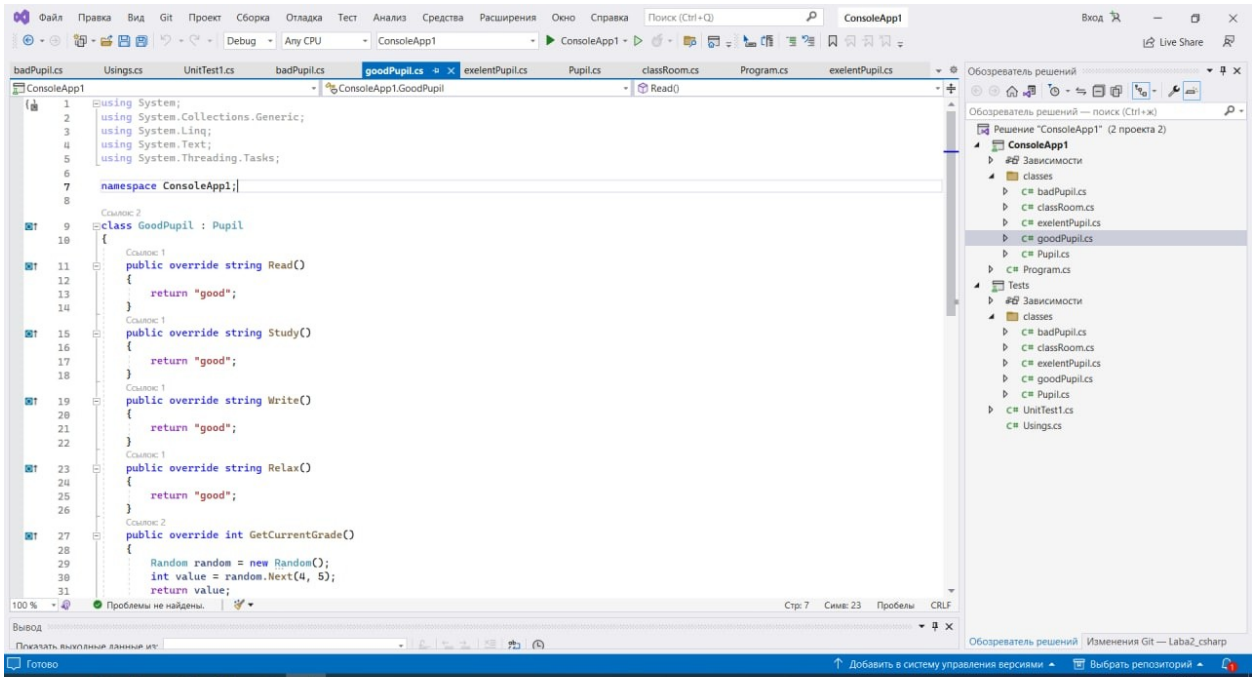


Рис. 5 – Класс GoodPupil

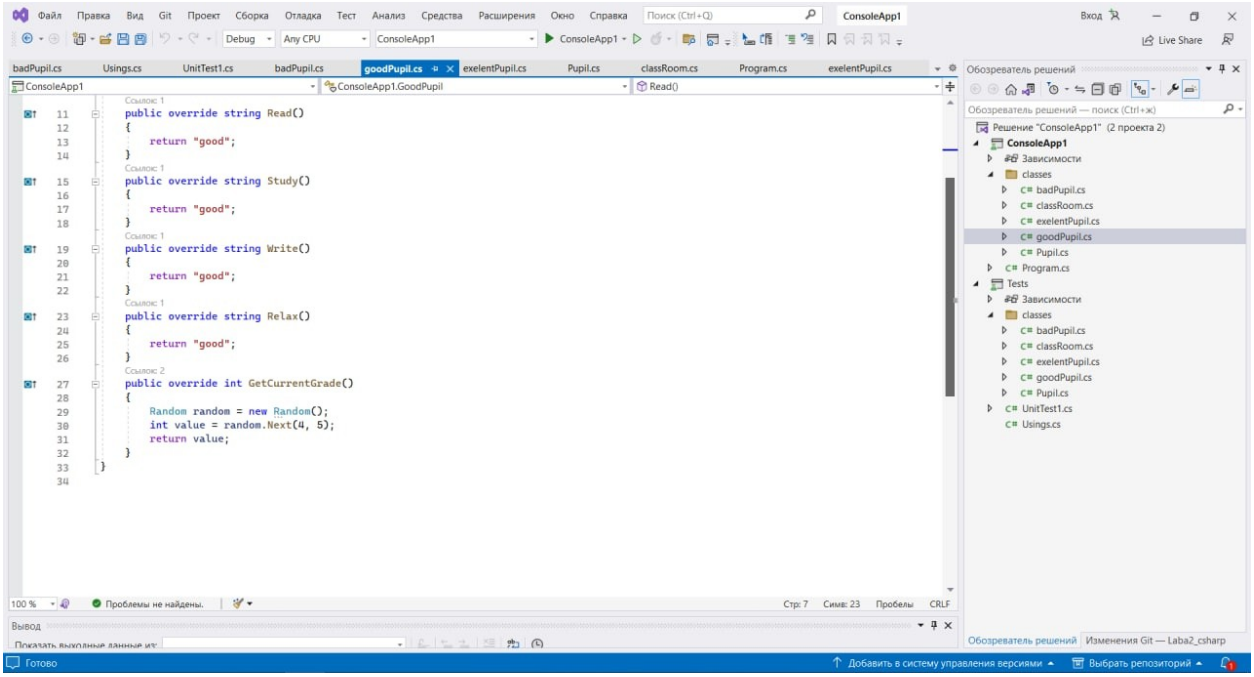


Рис. 6 – Класс GoodPupil

Реализация производного класса BadPupil:

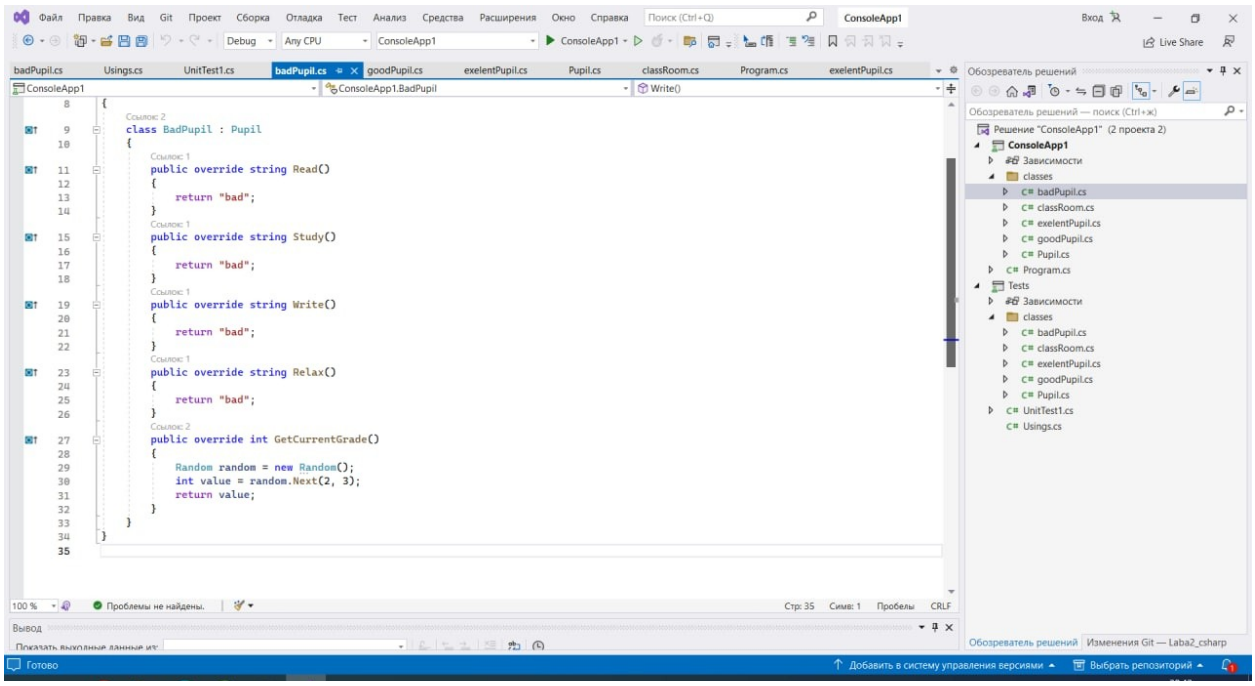


Рис. 7 – Класс BadPupil

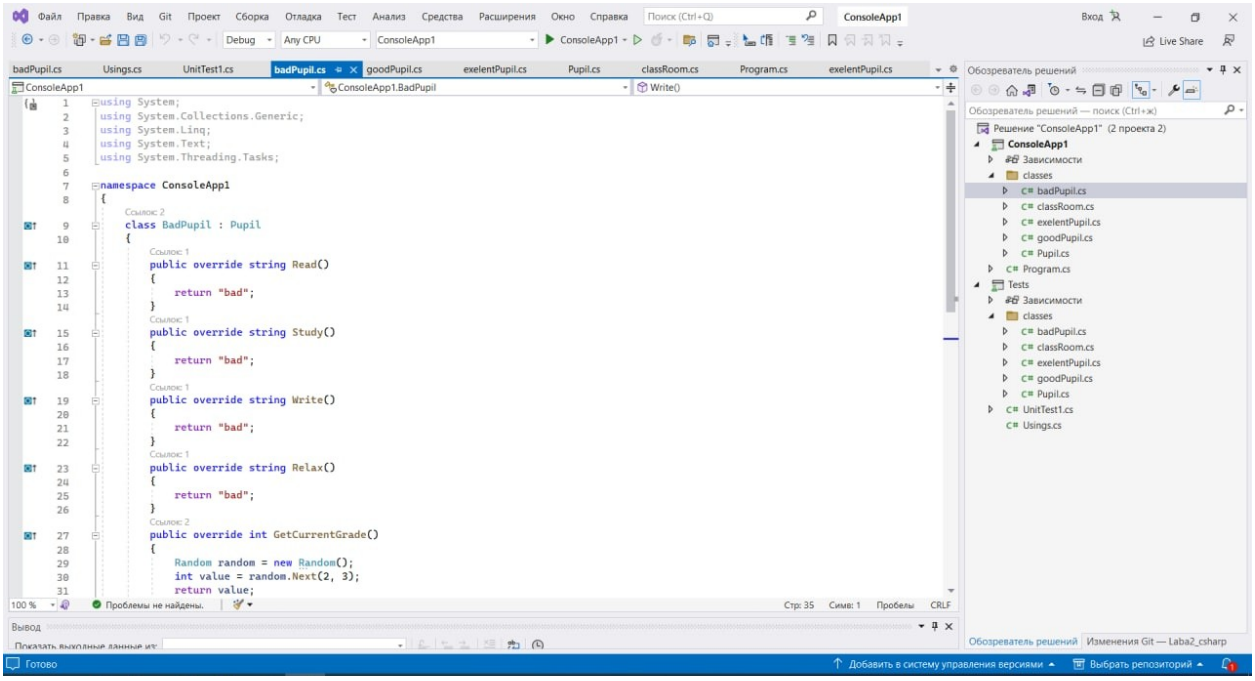


Рис. 8 – Класс BadPupil

Реализация производного класса ExcelentPupil:

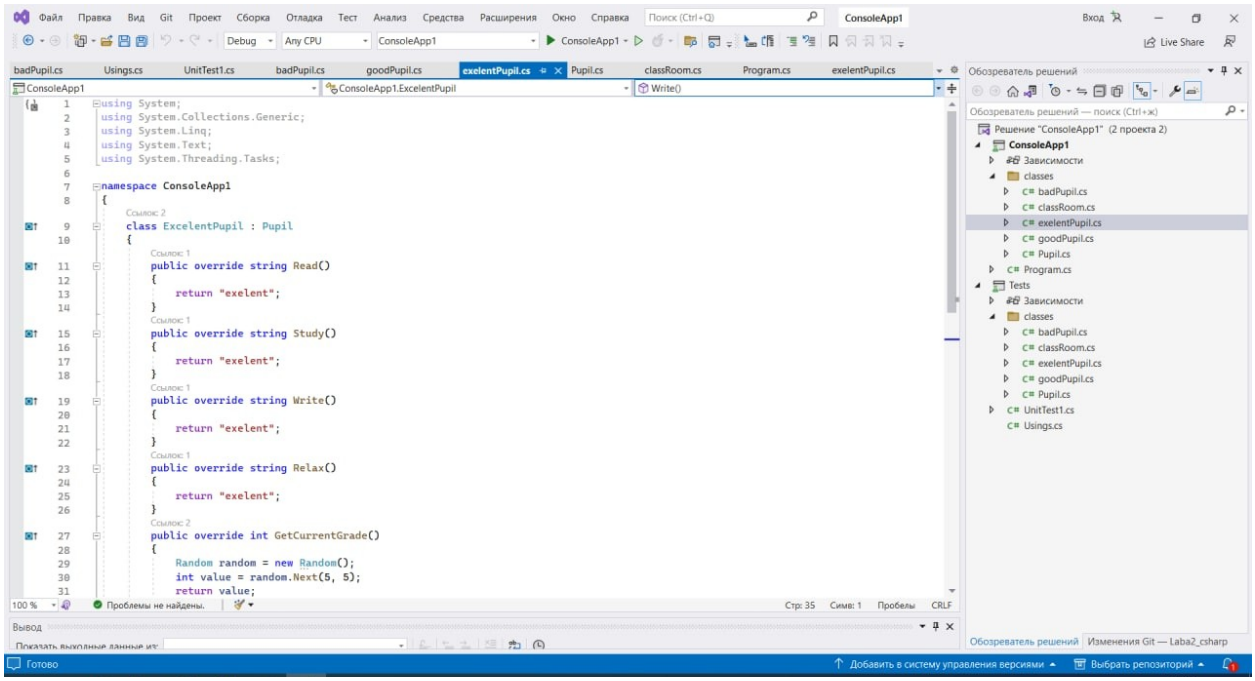


Рис. 9 – Класс ExcelentPupil

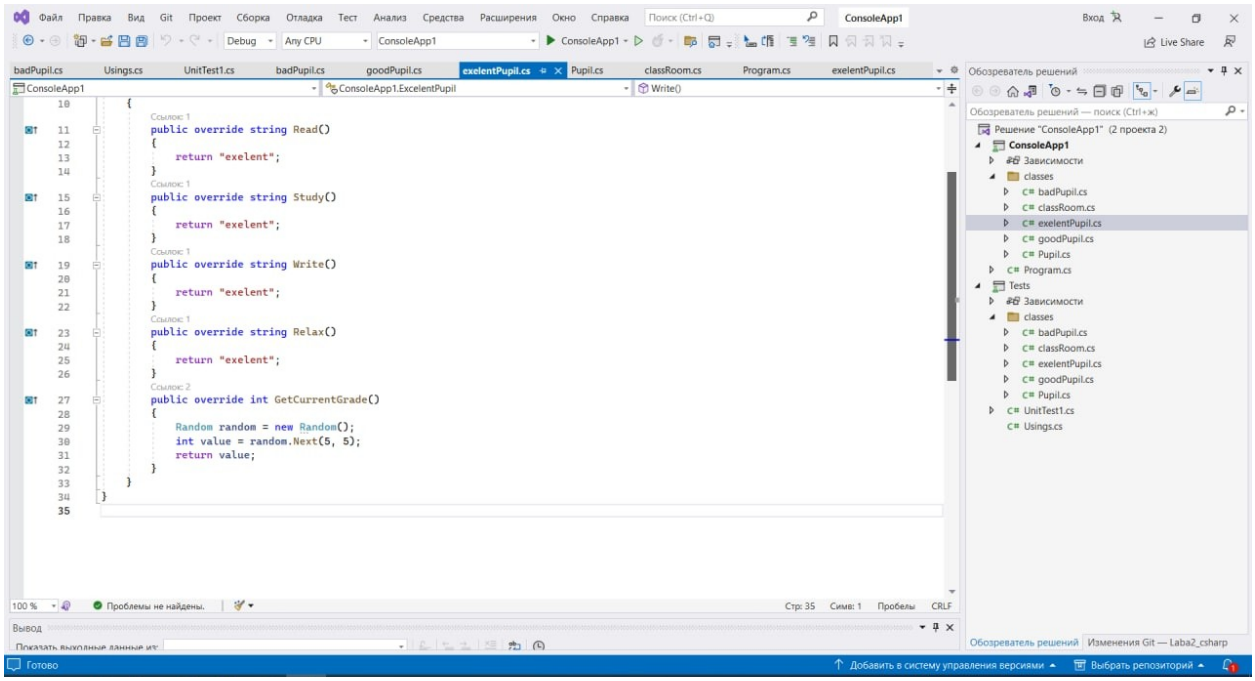


Рис. 10 – Класс ExcelentPupil

Unit – тесты:

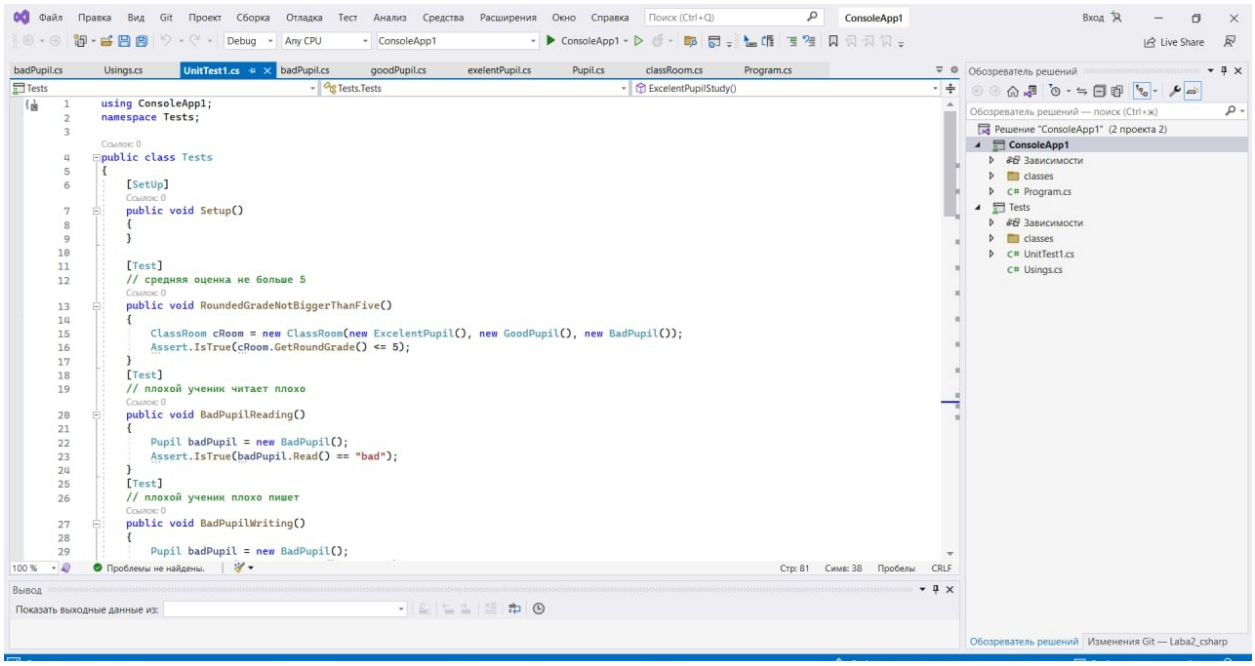


Рис. 11 – Unit – тесты

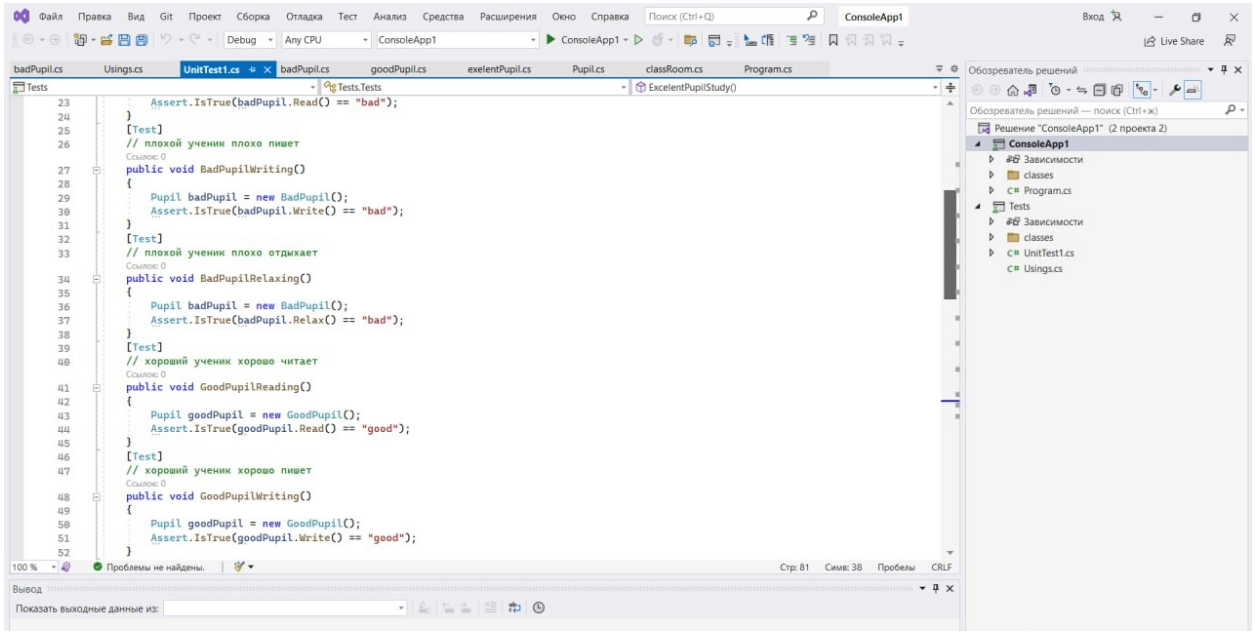


Рис.12 – Unit – тесты

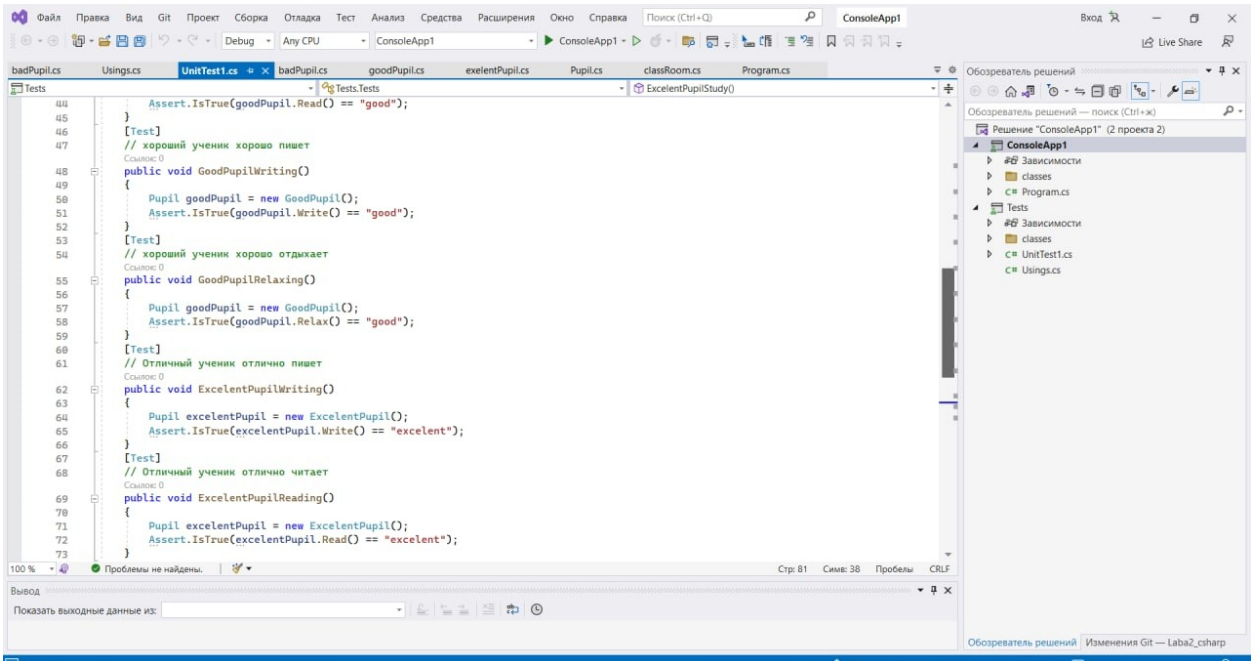


Рис. 13 - Unit – тесты

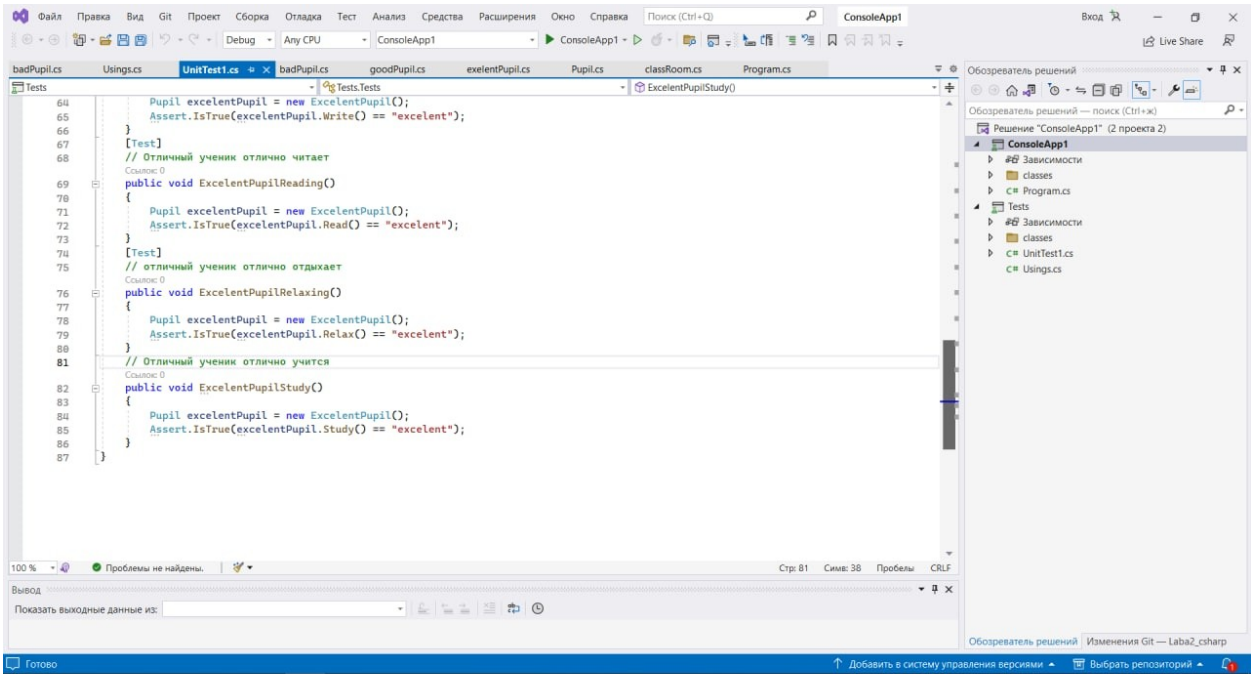


Рис. 14 - Unit – тесты