

**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:

```
1 using System;
2 using System.Collections;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8
9 namespace ConsoleApp1;
10 class Classroom
11 {
12     Stack<Pupil> innerClass = new Stack<Pupil>();
13     static int pupilsCount = 4;
14     public int GetRoundGrade()
15     {
16         int roundedGrade = 0;
17         foreach(Pupil pupil in innerClass)
18         {
19             roundedGrade += pupil.GetCurrentGrade();
20         }
21         roundedGrade /= pupilsCount;
22         return roundedGrade;
23     }
24 }
25 public Classroom(Pupil p1, Pupil p2)
26 {
27     innerClass.Push(p1);
28     innerClass.Push(p2);
29     innerClass.Push(new GoodPupil());
30     innerClass.Push(new GoodPupil());
31 }
32 public Classroom(Pupil p1, Pupil p2, Pupil p3)
33 {
34     innerClass.Push(p1);
35     innerClass.Push(p2);
36     innerClass.Push(p3);
37     innerClass.Push(new ExelentPupil());
38 }
39 }
40
41
42
```

Рис. 1 – Класс Classroom

```
18         roundedGrade += pupil.GetCurrentGrade();
19     }
20 }
21 roundedGrade /= pupilsCount;
22 return roundedGrade;
23 }
24 }
25 public Classroom(Pupil p1, Pupil p2)
26 {
27     innerClass.Push(p1);
28     innerClass.Push(p2);
29     innerClass.Push(new GoodPupil());
30     innerClass.Push(new GoodPupil());
31 }
32 public Classroom(Pupil p1, Pupil p2, Pupil p3)
33 {
34     innerClass.Push(p1);
35     innerClass.Push(p2);
36     innerClass.Push(p3);
37     innerClass.Push(new ExelentPupil());
38 }
39 }
40
41
42
```

Рис. 2 - Класс Classroom

Реализация базового класса Pupil:

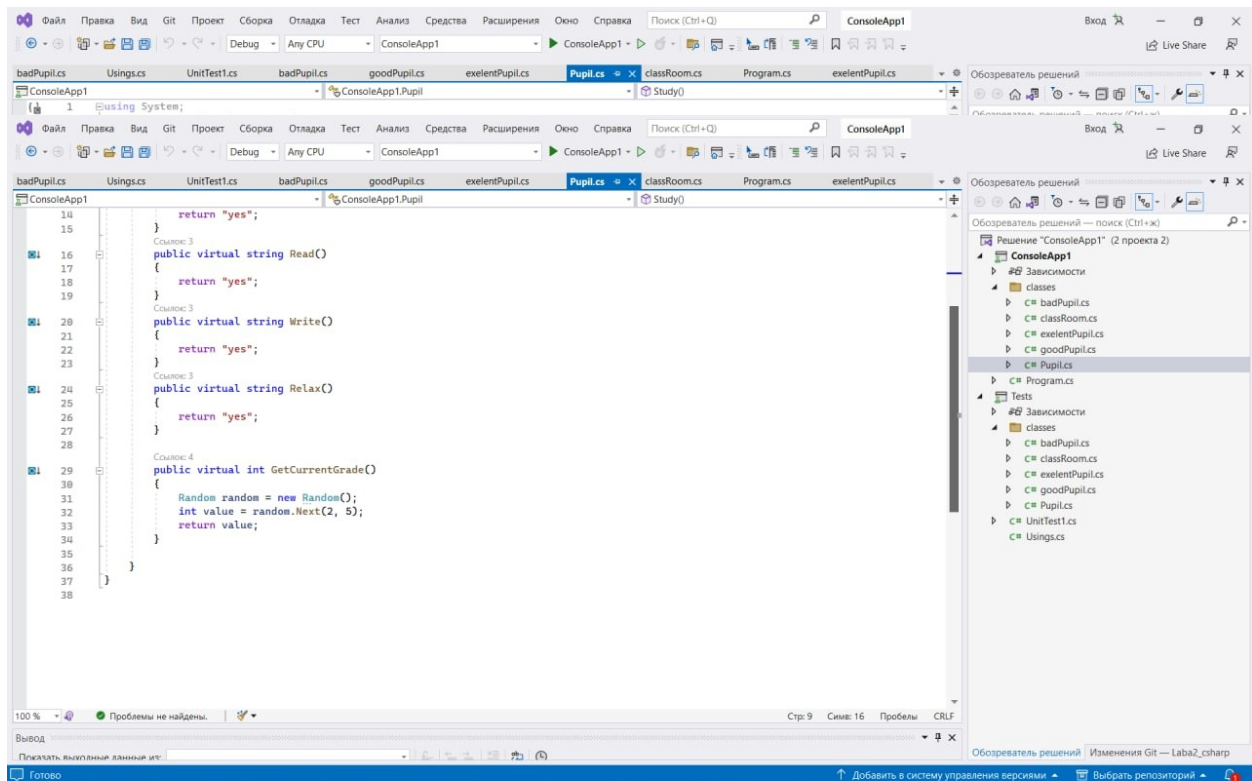


Рис.3 – Класс Pupil

Рис. 4 – Класс Pupil

Реализация производного класса GoodPupil:

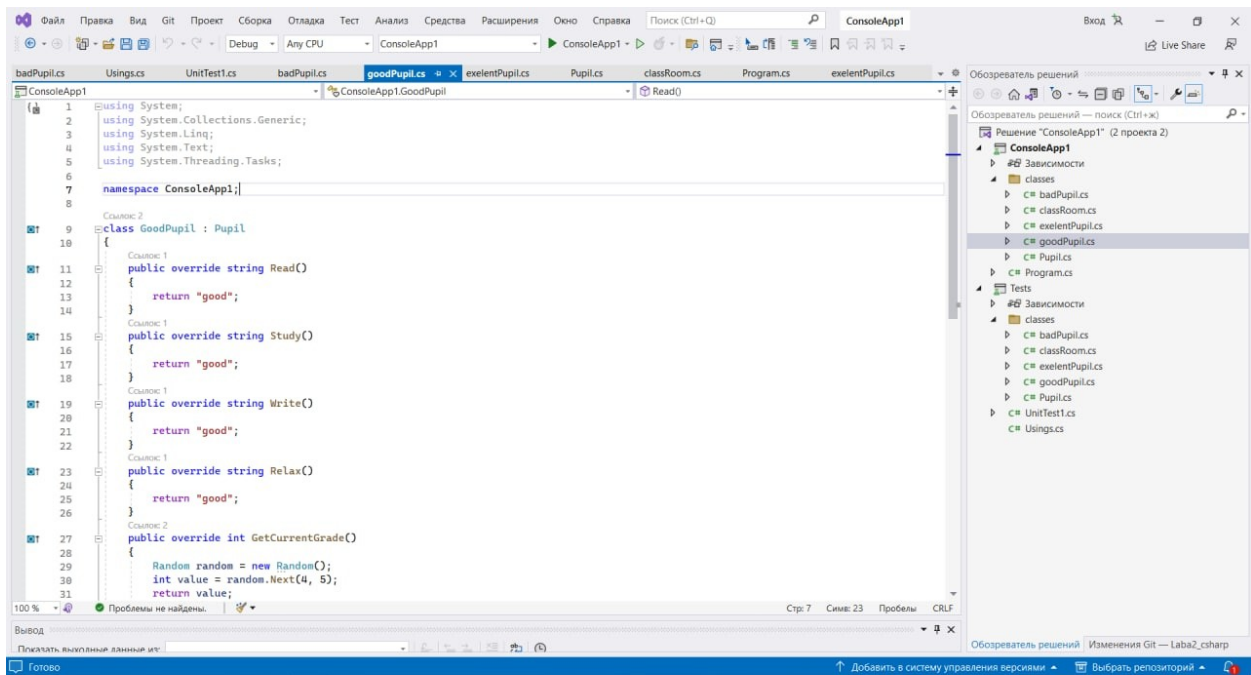


Рис. 5 – Класс GoodPupil

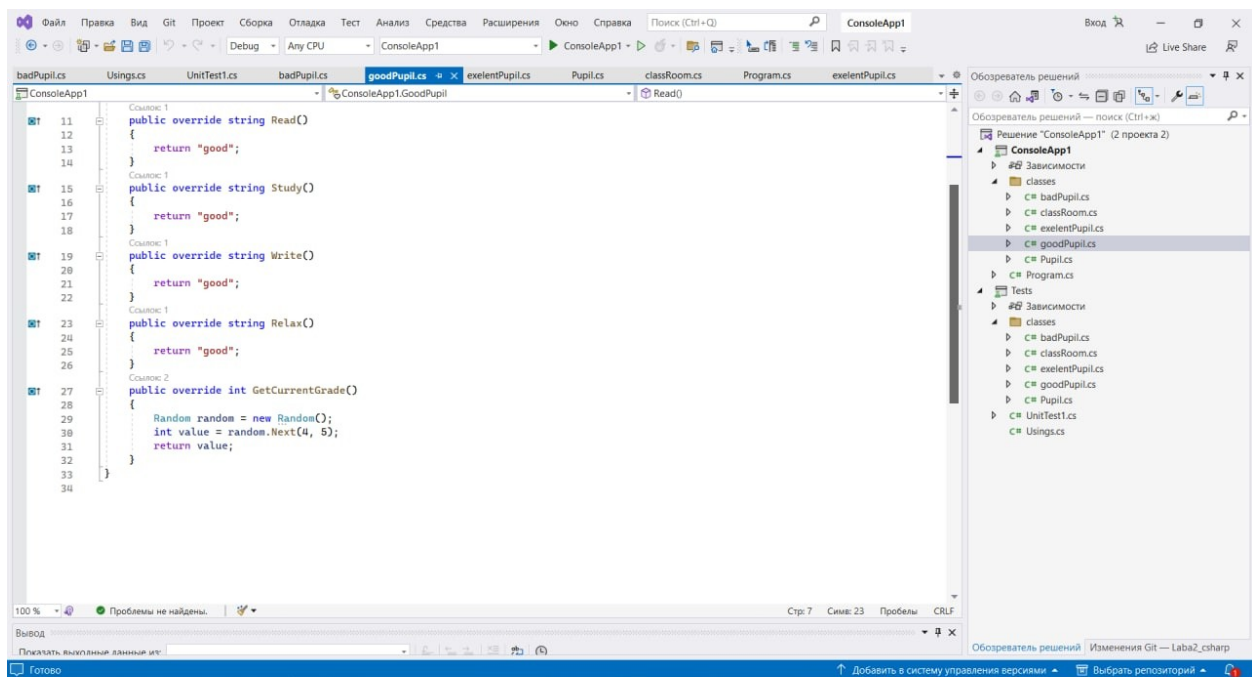


Рис. 6 – Класс GoodPupil

Реализация производного класса BadPupil:

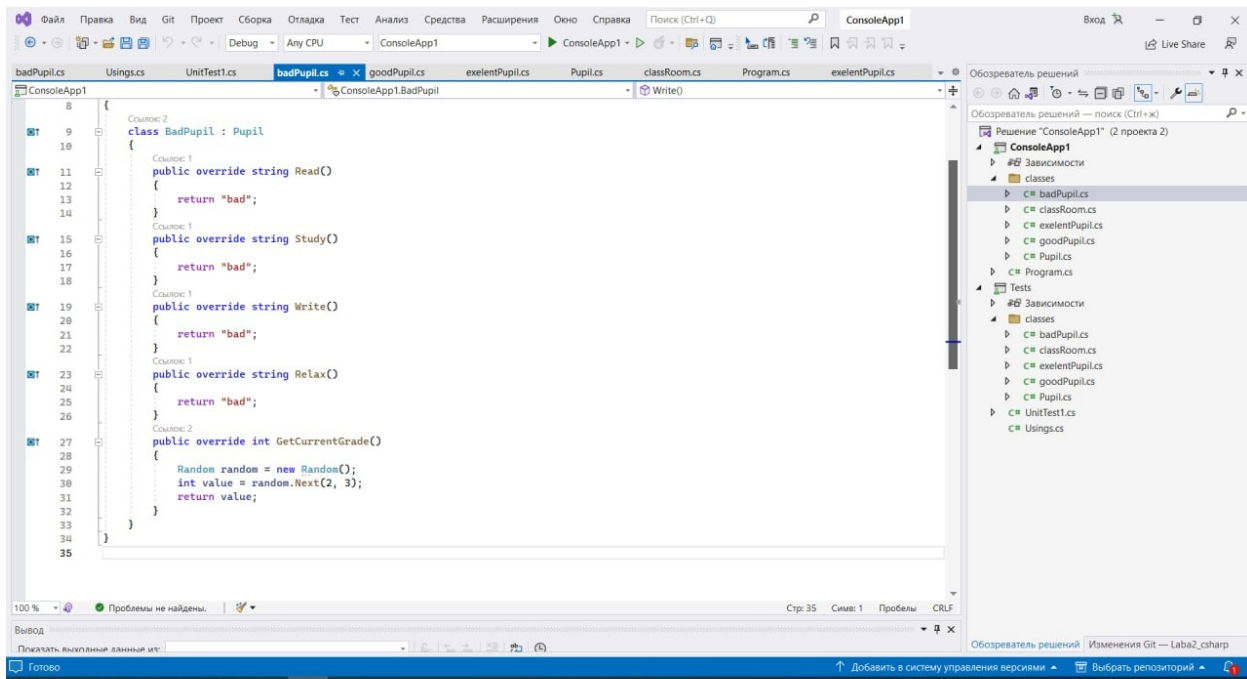


Рис. 7 – Класс BadPupil

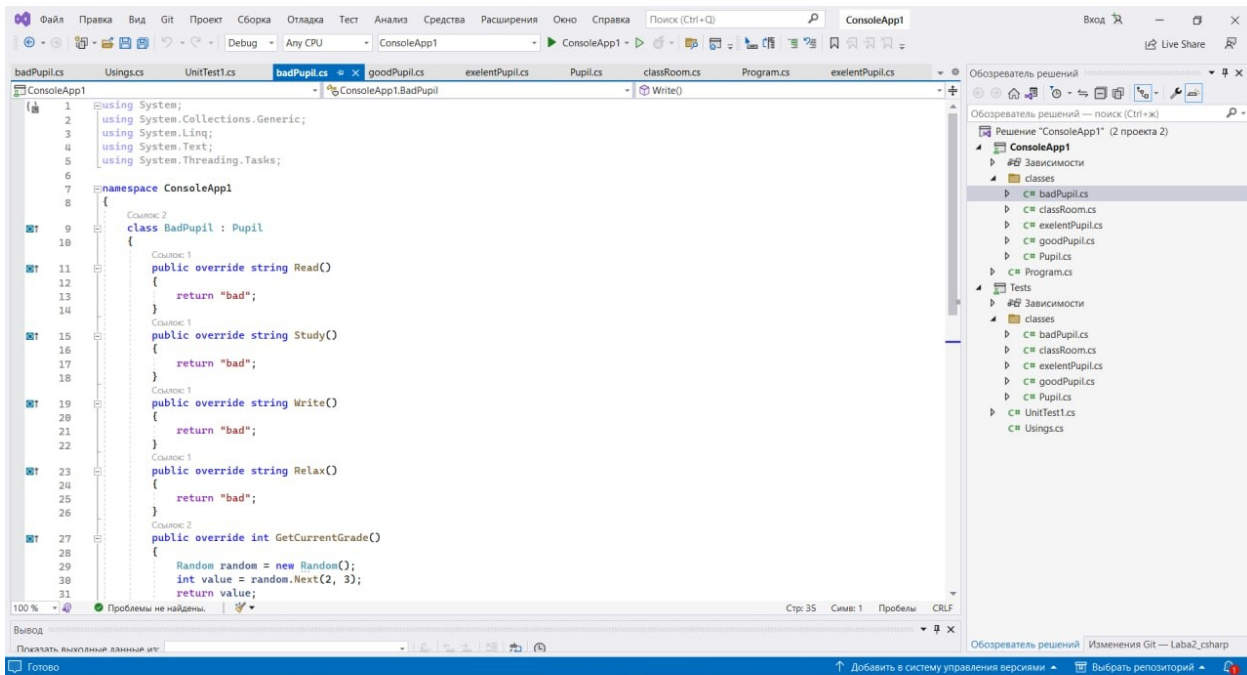


Рис. 8 – Класс BadPupil

Реализация производного класса ExcelentPupil:

The screenshot shows the Visual Studio IDE with the 'ConsoleApp1.ExcelentPupil' file open. The code defines a class 'ExcelentPupil' that inherits from 'Pupil'. It overrides the 'Read()', 'Study()', 'Write()', and 'Relax()' methods to return 'exelent', and the 'GetCurrentGrade()' method to return a random grade between 1 and 5. The 'Pupil' class is not visible in this view.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ConsoleApp1
8 {
9     class ExcelentPupil : Pupil
10    {
11        public override string Read()
12        {
13            return "exelent";
14        }
15        public override string Study()
16        {
17            return "exelent";
18        }
19        public override string Write()
20        {
21            return "exelent";
22        }
23        public override string Relax()
24        {
25            return "exelent";
26        }
27        public override int GetCurrentGrade()
28        {
29            Random random = new Random();
30            int value = random.Next(5, 5);
31            return value;
32        }
33    }
34 }
35
```

Рис. 9 – Класс ExcelentPupil

This screenshot is identical to the previous one, showing the implementation of the 'ExcelentPupil' class in Visual Studio. The code is the same as in Figure 9.

```
10
11        public override string Read()
12        {
13            return "exelent";
14        }
15        public override string Study()
16        {
17            return "exelent";
18        }
19        public override string Write()
20        {
21            return "exelent";
22        }
23        public override string Relax()
24        {
25            return "exelent";
26        }
27        public override int GetCurrentGrade()
28        {
29            Random random = new Random();
30            int value = random.Next(5, 5);
31            return value;
32        }
33    }
34 }
35
```

Рис. 10 – Класс ExcelentPupil

Unit – тесты:

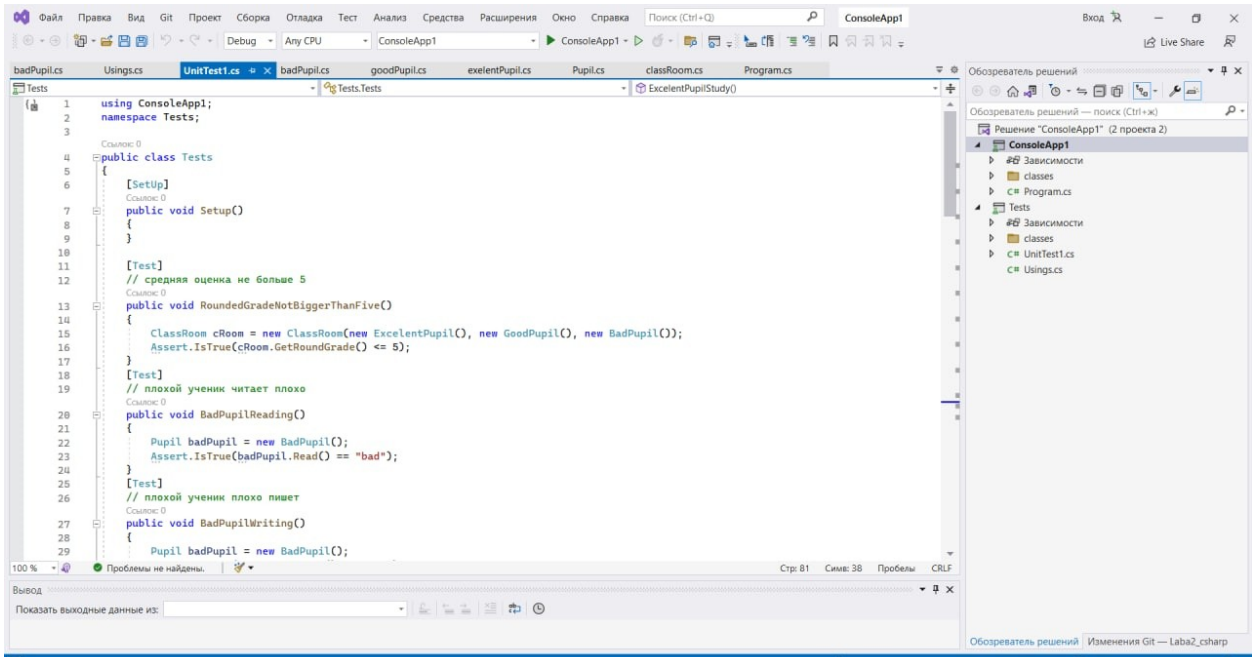


Рис. 11 – Unit – тесты

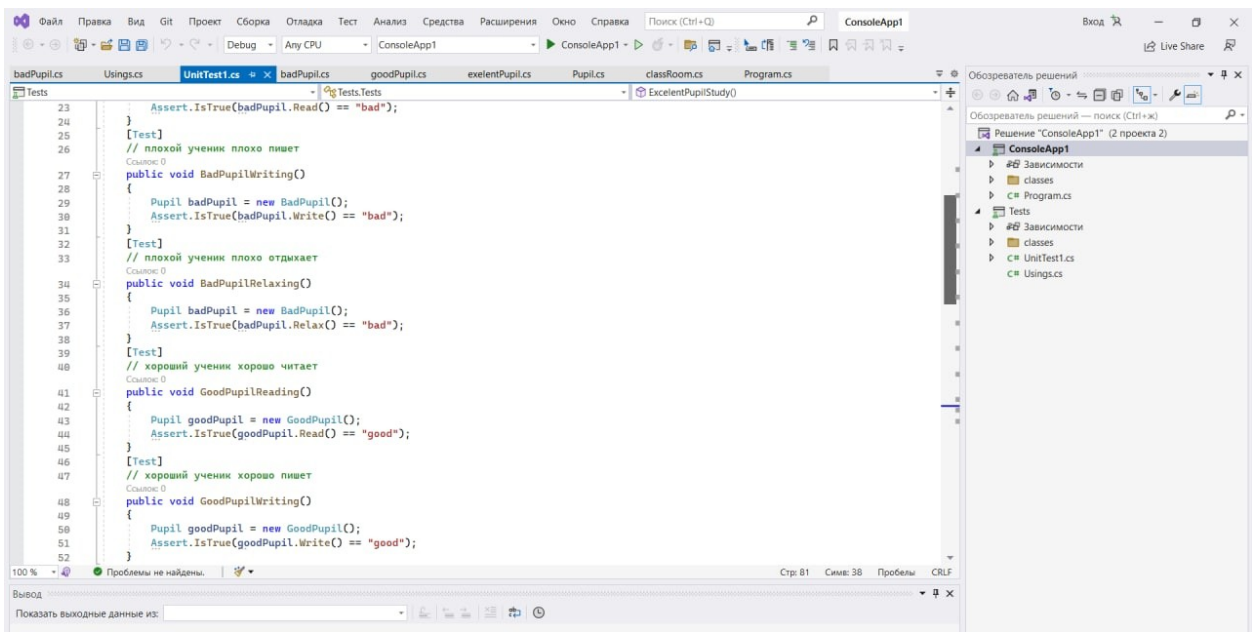


Рис.12 – Unit – тесты

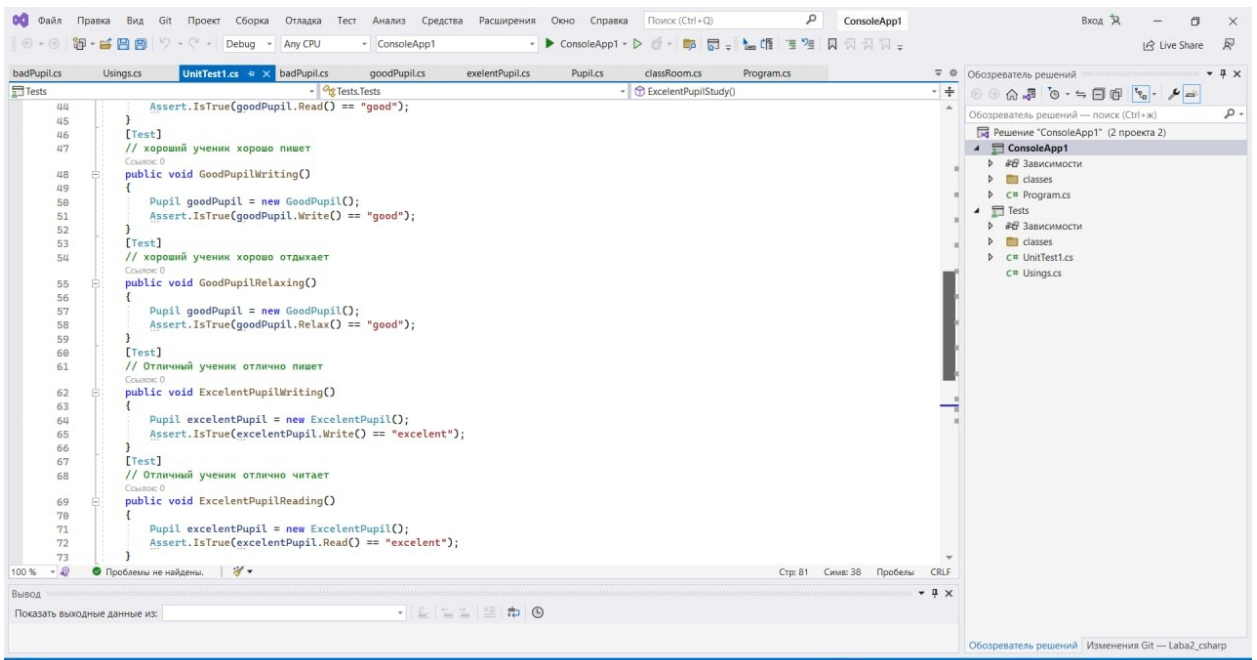


Рис. 13 - Unit – тесты

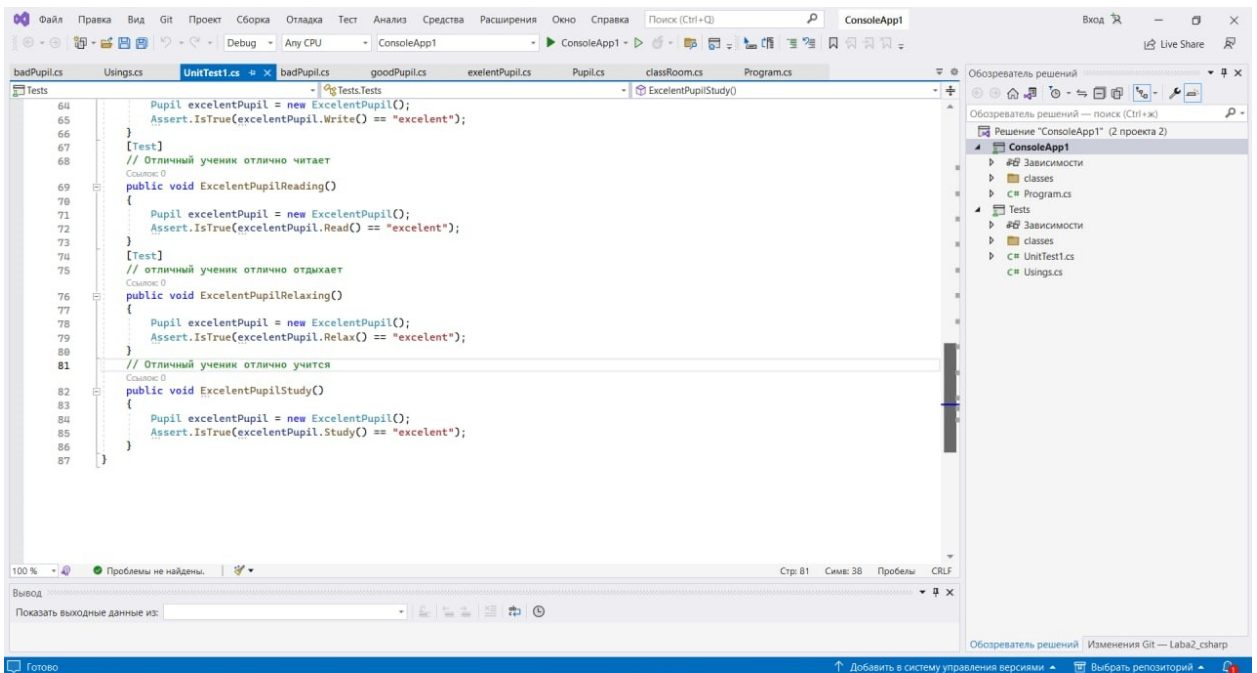


Рис. 14 - Unit – тесты