**Практическая работа № 1**

**«Тестирование структуры программных модулей»**

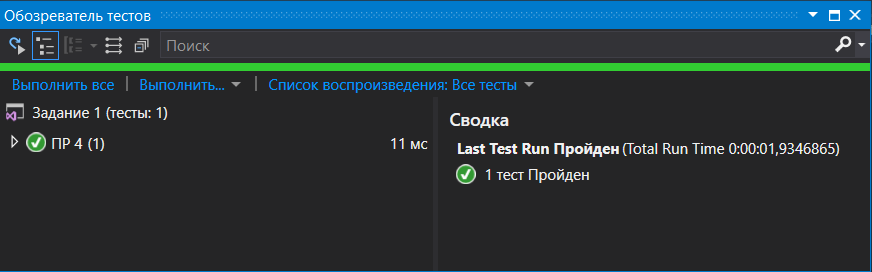
**Цель работы:** закрепление практических навыков по проведению модульного

тестирования.

*Создаём тесты:*

*1.*

Начинается сборка и запуск теста, после появится обозреватель тестов.



Исходный код программы:

static void Main(string[] args)

{

Console.Write("Введите строку: ");

string s = Console.ReadLine(); //Ввод строки

Regex r = new Regex(@"\b(\w){3}\b", RegexOptions.IgnoreCase

Match words = r.Match(s);

while (words.Success)

{

Console.WriteLine(words);

words = words.NextMatch();

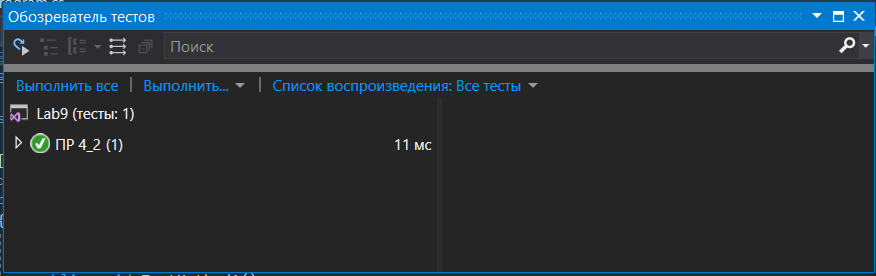
}

Console.ReadKey();

}

*2.*

Начинается сборка и запуск теста, после появится обозреватель тестов.



Исходный код:

Console.WriteLine("Введите x1");

double x1 = double.Parse(Console.ReadLine());

Console.WriteLine("Введите y1");

double y1 = double.Parse(Console.ReadLine());

Console.WriteLine("Введите x2");

double x2 = double.Parse(Console.ReadLine());

Console.WriteLine("Введите y2");

double y2 = double.Parse(Console.ReadLine());

double rastoyanie = Math.Sqrt(Math.Pow(x2 - x1,2) + Math.Pow(y2 - y1,2));

Console.WriteLine("Расстояние между двумя точками= {0}", rastoyanie);

Console.ReadKey();

При создании тестов были использованы методы класса Assert и СollectionAssert.

Классы Assert и СollectionAssert содержат несколько статических методов, которые можно использовать для проверки результатов в тестовых методах.

*Выводы:*

На основе проведённых тестов стала понятна корректность работы модулей исходного кода. В первом случае исходный код был полностью корректен, во втором случае были намеренно допущены ошибки, по результатам теста сразу стало понятно какой из модулей неисправен.

1. Назовите преимущества и недостатки модульного тестирования.

*Преимущества*

Модульное тестирование имеет ряд преимуществ, таких как:

• Это помогает в разработке и рефакторинге лучшего кода

• Гарантирует обеспечение гарантированного качества сборки, когда она интегрируется со сборкой

• Дефекты в приложении выявляются и устраняются на ранней стадии, это снижает стоимость тестирования.

*Недостатки*

• Нельзя ожидать, что модульное тестирование выявит каждую ошибку в программе.

• Невозможно оценить все пути выполнения даже в самых тривиальных программах

• Модульное тестирование по своей природе ориентировано на единицу кода. Следовательно, он не может отловить ошибки интеграции или ошибки системного уровня

2. Что такое unit –тест?

Юнит-тест (unit test), или модульный тест, — это программа, которая проверяет работу небольшой части кода.