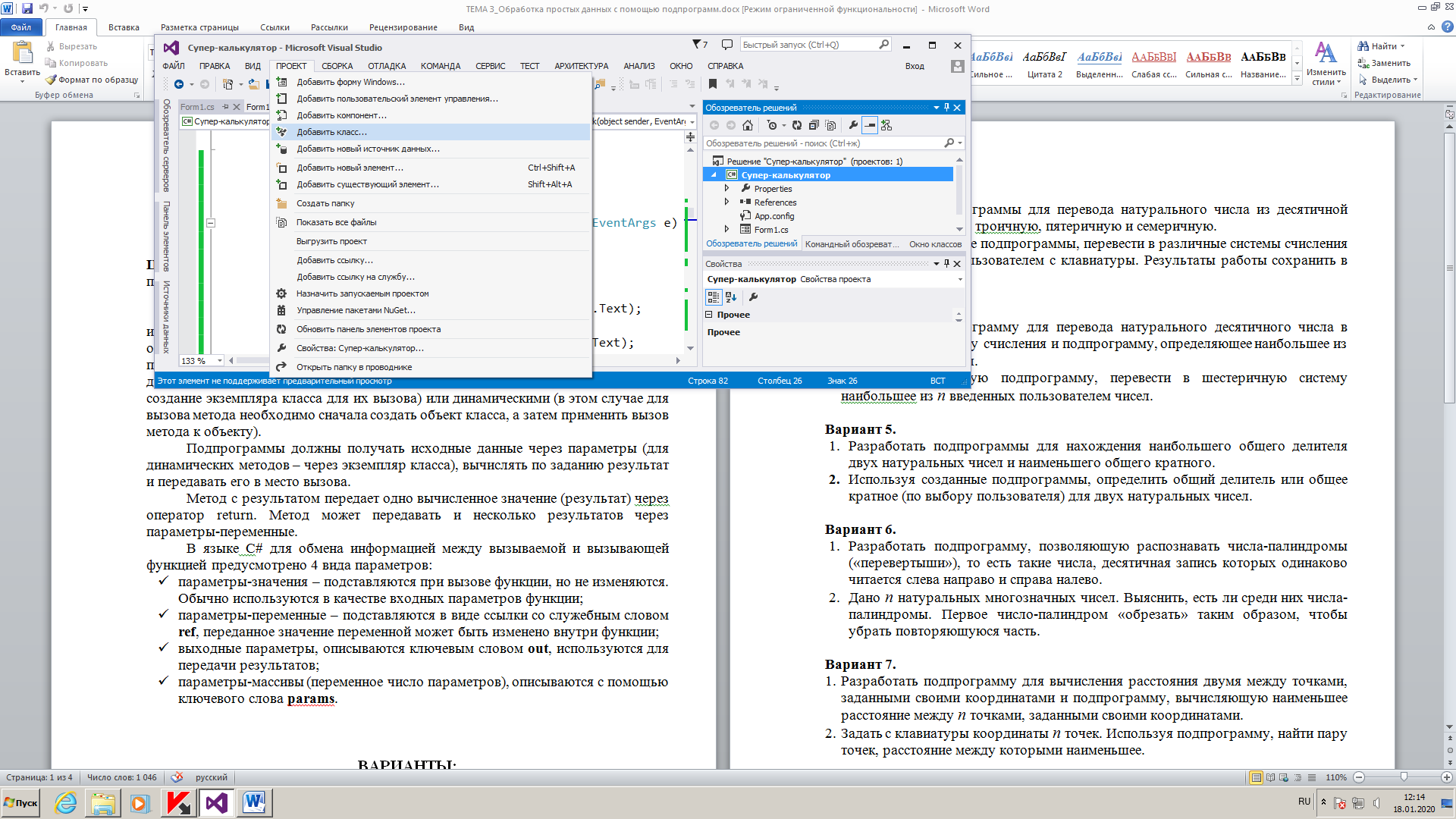
**Тема 3:** Использование подпрограмм для обработки

простых типов данных

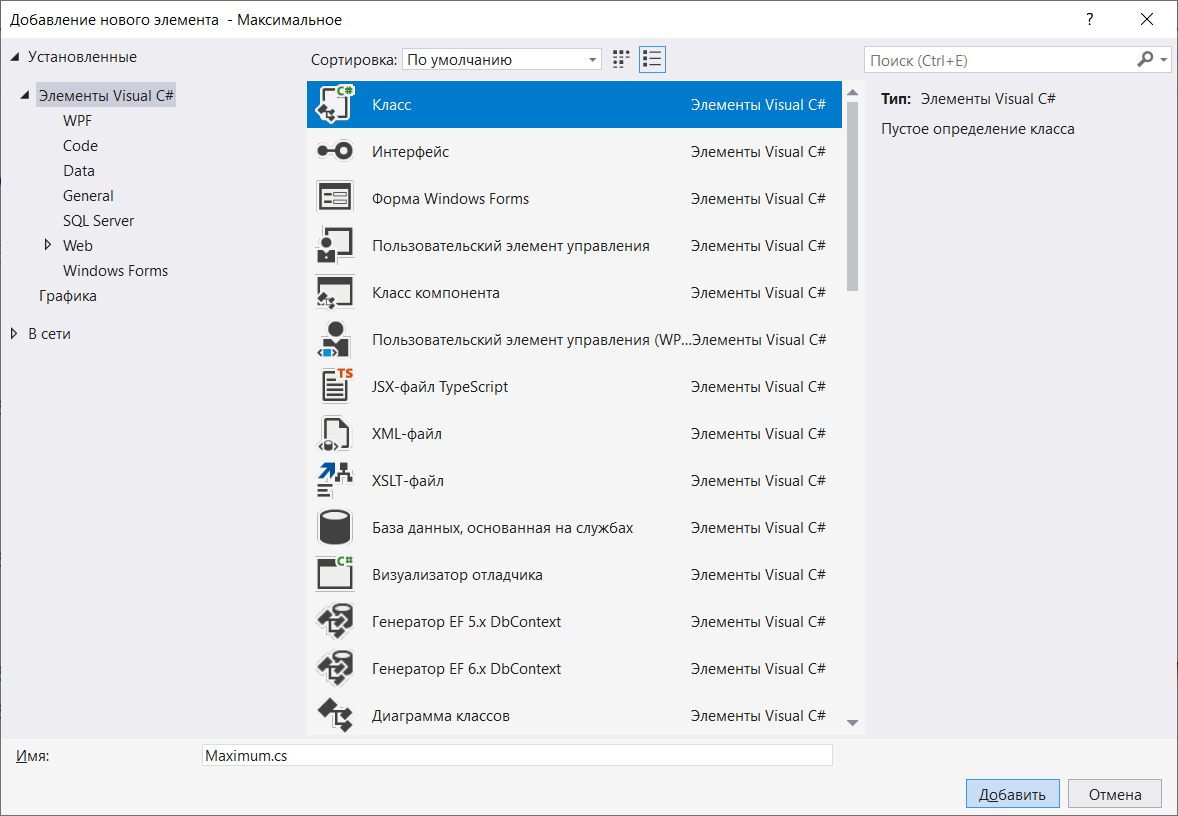
**Цель работы:** Получить практические навыки разработки и использования подпрограмм для обработки простых типов данных.

**Задание 1.** Создание дополнительного класса и разработка подпрограмм (методов) класса

Подпрограммы расположить в отдельном файле (модуле). Чтобы добавить в проект новый модуль, в обозревателе решений выделите проект и выберите пункт меню **Проект→Добавить класс.** Введите название своего класса.



*Рисунок 1 Добавление нового класса в проект*



*Рисунок 2 Ввод названия нового класса*

***Статические и динамические методы.*** Методы класса должны иметь уровень доступа public, так как предполагается, что они будут вызываться из другого класса (класса формы). Методы могут быть статическими — в этом случае не требуется создание экземпляра класса для их вызова, метод вызывается через название класса, или динамическими — в этом случае для вызова метода необходимо сначала создать объект класса, а затем применить вызов метода к объекту.

***Как метод получает исходные данные?*** Методы должны получать исходные данные через параметры (для динамических методов – в основном через данные самого экземпляра класса).

В языке С# для обмена информацией между вызываемой и вызывающей функцией предусмотрено **4 вида параметров:**

1. *Параметры-значения* – подставляются при вызове функции, но не изменяются. Обычно используются в качестве входных параметров функции;
2. *Параметры-переменные* – подставляются в виде ссылки со служебным словом **ref**, переданное значение переменной может быть изменено внутри функции;
3. *Выходные параметры* – описываются ключевым словом **out**, используются для передачи результатов;
4. *Параметры-массивы* (переменное число параметров), описываются с помощью ключевого слова **params**.

***Как метод передает результаты?*** Метод долженвычислять результат и передавать его в место вызова (именно передавать, а не выводить на экран!)

Метод с результатом передает только одно вычисленное значение (результат) через оператор return. В случае необходимости методы могут передавать несколько результатов через параметры-переменные и выходные параметры.

***Как метод сигнализирует об ошибках?*** Если подпрограмма не может сформировать результат по каким-то причинам (например, при ошибочных исходных данных), то сообщение об ошибке в теле метода тоже не печатается, а производиться генерация исключения.

*Пример класса со статическими методами:*

class Maximum

{

/// <summary>

/// Вычисление максимального значения из двух целых чисел

/// </summary>

/// <param name="a">первый аргумент</param>

/// <param name="b">второй аргумент</param>

/// <returns>максимальное из двух целочисленных аргументов</returns>

public static int Max(int a, int b)

{

if (a > b) return a;

else

if (b > a) return b;

else

throw new Exception("Числа равны!");

}

/// <summary>

/// Вычисление максимального значения из трёх целых чисел

/// </summary>

/// <param name="a">первый аргумент</param>

/// <param name="b">второй аргумент</param>

/// <param name="c">третий аргумент</param>

/// <returns>максимальное из трёх целочисленных аргументов</returns>

public static int Max(int a, int b, int c)

{

if (a == b && b == c)

throw new Exception("Числа равны!");

int max = a;

if (b > max)

max = b;

if (c > max)

max = c;

return max;

}

/// <summary>

/// Вычисление максимального значения в целочисленном массиве

/// </summary>

/// <param name="x">на вход подаётся целочисленный массив</param>

/// <returns>значение максимального элемента массива</returns>

public static int Max(int[] x)

{

int max = x[0];

for (int i = 0; i < x.Length; i++)

{

if (x[i] > max)

max = x[i];

}

int k = 0;

for (int i = 0; i < x.Length; i++)

{

if (x[i] == max)

k++;

}

if (k == x.Length)

throw new Exception("Все элементы массива равны!");

return max;

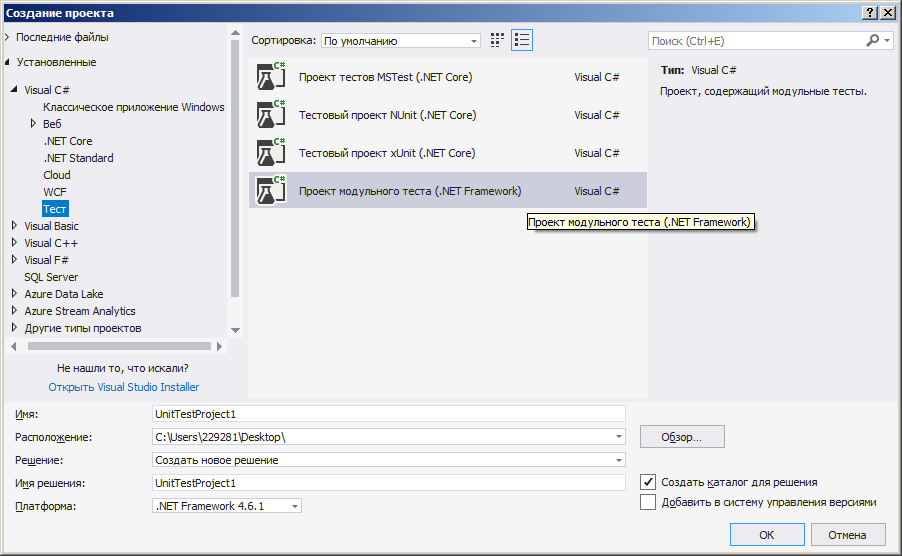
}

}

**Задание 2.** Тестирование подпрограмм с помощью тест-кейсов

<https://docs.microsoft.com/ru-ru/visualstudio/test/walkthrough-creating-and-running-unit-tests-for-managed-code?view=vs-2017>

* Для тестирования созданного класса создать проект UnitTestsProject.   
  При создании тестового проекта добавляйте его в существующее решение.
* В тестовом проекте создать 2-4 функции тестирования
* Запустить юнит-тесты и продемонстрировать успешное и неуспешное тестирование



*Рисунок 3 Создание проекта юнит-тестов*

**Примеры юнит-тестов:**

[TestClass]

public class UnitTest1

{

[TestMethod]

public void TestMethodMax()

{

// исходные данные для теста

int first = 16;

int second = 10;

// ожидаемый результат

int experted = 16;

//вызов тестируемой функции

int actual = Maximum.Max(first, second);

// сравнение результатов

Assert.AreEqual(experted, actual, 0.0, "Ожидаемый результат не получен!");

}

[TestMethod]

public void TestMethodMax3()

{

// исходные данные для теста

int first = 16;

int second = 10;

int thread = 34;

// ожидаемый результат

int experted = 34;

//вызов тестируемой функции

int actual = Maximum.Max(first, second, thread);

// сравнение результатов

Assert.AreEqual(experted, actual, 0.0, "Ожидаемый результат не получен!");

}

[TestMethod]

public void TestMethodMaxException()

{

// исходные данные для теста

int first = 16;

int second = 16;

try

{

//вызов тестируемой функции

int actual = Maximum.Max(first, second);

}

catch (Exception e)

{

StringAssert.Contains(e.Message, "Числа равны.");

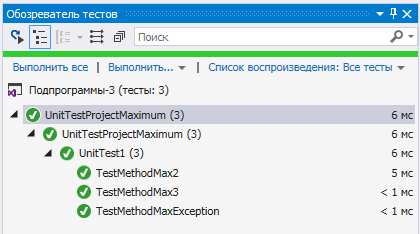
return;

}

Assert.Fail("Ожидаемое исключение не получено.");

}

}



*Рисунок 3 Успешное выполнение юнит-тестов*

**Задание 3.** Создание пользовательского интерфейса (модуля формы)

Необходимо настроить внешний вид и поведение компонентов формы и создать методы обработки событий. Модуль формы должен содержать код, который продемонстрирует вызов методов из дополнительного класса по вашему заданию.

**ВАРИАНТЫ:**

**Вариант 1.**

1. Разработать подпрограммы, позволяющие распознавать и складывать троичные числа.
2. Используя созданные подпрограммы, определить сумму двух троичных чисел, если они правильно введены пользователем.

**Вариант 2.**

1. Разработать подпрограмму, позволяющую распознавать и увеличивать на 1 пятеричные числа.
2. Используя созданную подпрограмму, ввести с клавиатуры несколько чисел, и те из них, которые являются пятеричными, увеличить на 1.

**Вариант 3.**

1. Разработать подпрограммы для перевода натурального числа из десятичной системы счисления в троичную, пятеричную и семеричную.
2. Используя созданные подпрограммы, перевести в различные системы счисления числа, вводимые пользователем с клавиатуры. Результаты работы сохранить в текстовый файл.

**Вариант 4.**

1. Разработать подпрограмму для перевода натурального десятичного числа в шестеричную систему счисления и подпрограмму, определяющее наибольшее из *n* натуральных чисел.
2. Используя созданную подпрограмму, перевести в шестеричную систему наибольшее из *n* введенных пользователем чисел.

**Вариант 5.**

1. Разработать подпрограммы для нахождения наибольшего общего делителя двух натуральных чисел и наименьшего общего кратного.
2. Используя созданные подпрограммы, определить общий делитель или общее кратное (по выбору пользователя) для двух натуральных чисел.

**Вариант 6.**

1. Разработать подпрограмму, позволяющую распознавать числа-палиндромы («перевертыши»), то есть такие числа, десятичная запись которых одинаково читается слева направо и справа налево.
2. Дано *n* натуральных многозначных чисел. Выяснить, есть ли среди них числа-палиндромы. Первое число-палиндром «обрезать» таким образом, чтобы убрать повторяющуюся часть.

**Вариант 7.**

1. Разработать подпрограмму для вычисления расстояния двумя между точками, заданными своими координатами и подпрограмму, вычисляющую наименьшее расстояние между *n* точками, заданными своими координатами.
2. Задать с клавиатуры координаты *n* точек. Используя подпрограмму, найти пару точек, расстояние между которыми наименьшее.

**Вариант 8.**

1. Разработать подпрограмму для вычисления расстояния двумя между точками, заданными своими координатами и подпрограмму, вычисляющую периметр выпуклого многоугольника, заданного координатами своих вершин.
2. Вычислить периметр многоугольника, заданного координатами своих вершин, используя созданную подпрограмму. Координаты вершин прочитать из текстового файла.

**Вариант 9.**

1. Разработать подпрограмму для определения можно ли построить треугольник, заданный длинами своих сторон, и подпрограмму для вычисления периметра и площади треугольника по трем сторонам.
2. Используя созданную подпрограмму, вычислить площади всех треугольников, которые возможно построить, выбирая значения сторон из пяти заданных отрезков.

**Вариант 10.**

1. Разработать подпрограмму для вычисления корней квадратного уравнения в действительной области.
2. Используя созданную подпрограмму, вычислить корни всех квадратных уравнений, коэффициенты которых прочитаны из текстового файла.

**Вариант 11.**

1. Разработать подпрограмму, позволяющую распознавать простые числа и подпрограмму для нахождения всех простых чисел в заданном интервале от   
   *a* до *b*.
2. Используя созданную подпрограмму, найти и вывести на экран все простые числа из заданного интервала.

**Вариант 12.**

1. Разработать подпрограммы, вычисляющие сумму и произведение цифр натурального многозначного числа (с любым количеством разрядов).
2. Даны несколько натуральных многозначных чисел. Используя созданные подпрограммы, определить, в каком из них больше сумма цифр, и в каком из них больше произведение цифр. Числа прочитать из текстового файла.

**Вариант 13.**

1. Разработать подпрограмму, вычисляющую цифровой корень натурального числа. Для получения цифрового корня многозначного числа нужно сложить все цифры этого числа, затем все цифры найденной суммы и так далее, пока не получится однозначное число. Проверить работу подпрограммы, организовав её вызов с фактическими параметрами.
2. Запросить у пользователя день, месяц и год рождения, и записав все эти данные в виде единого многозначного числа, вычислить его цифровой корень с помощью созданной подпрограммы, и выдать сообщение: «Ваша счастливая цифра …»

**Вариант 14.**

1. Разработать подпрограмму для перевода натурального числа в пятеричную систему счисления.
2. Используя созданную подпрограмму, перевести в пятеричную систему счисления все числа, прочитанные из текстового файла.

**Вариант 15.**

1. Разработать подпрограмму для перевода натурального числа в шестеричную систему счисления.
2. Используя созданную подпрограмму, перевести в шестиричную систему счисления все числа, прочитанные из текстового файла.

**Вариант 16.**

1. Разработать подпрограмму для получения «перевернутого» натурального числа, то есть числа, цифры которого идут в обратном порядке, и подпрограмму, определяющую минимум среди *n* чисел.
2. Используя созданную подпрограмму, получить для каждого из *n* заданных натуральных чисел сумму самого числа и «перевернутого» числа, а также найти минимальную из этих сумм.

**Вариант 17.**

1. Разработать подпрограммы для расчета периметра и площади параллелограмма и равнобедренной трапеции.
2. Выбрать вид фигуры, запросить её измерения и, используя созданные подпрограммы, вычислить периметр и площадь фигуры.

**Вариант 18.**

1. Разработать подпрограмму определения, является ли заданный год високосным. Високосным считается год, значение которого делиться на 4, кроме тех годов, которые делятся на 100, но не делятся на 400. Разработать вторую подпрограмму для получения перечня всех високосных годов в заданном интервале.
2. Используя созданную подпрограмму, вывести все високосные года из указанного промежутка.

**Вариант 19.**

1. Разработать подпрограммы для расчета площади поверхности и объема цилиндра и конуса, если известны радиус основания и высота фигур.
2. Выбрать вид фигуры, запросить её измерения и, используя созданные подпрограммы, вычислить площади поверхности и объем определенной фигуры.

**Вариант 20.**

1. Разработать подпрограмму, которая позволяет складывать и вычитать комплексные числа, а также выводить на экран комплексное число в стандартном виде.
2. Организовать пользовательский интерфейс, позволяющий ввести два комплексных числа (то есть их действительную и мнимую часть), вывести эти числа в стандартном виде, а также вывести их сумму и разность также в стандартном виде.