# Лабораторная работа № 4

# МЕТОДЫ

**Цель:** познакомиться с описанием и использованием пользовательских методов в программе

**МЕТОДЫ И ПАРАМЕТРЫ**

При разработке большинства приложений их разделяют на функциональные модули. Это позволяет несколько раз использовать одни и те же участки кода для других приложений.

В С# приложение состоит из классов, которые содержат именованные блоки кода, называемые методами. Метод – это член класса, который может осуществлять действия или вычислять значения.

## Использование методов

В С# программа состоит из классов, содержащих методы. Метод может выполнять действие или вычислять значение.

Метод – это набор операторов, которые выполняются вместе. В С# все методы принадлежат какому-либо классу. Для создания метода необходимо задать его имя, определить список параметров и тело метода.

Для вызова метода используется имя метода, если у метода есть параметры, то необходимо их определить. Для вызова метода другого класса необходимо, чтобы он был объявлен как public, вызов осуществляется по имени класса и метода.

Общее определение методов выглядит следующим образом:

Изображение выглядит как текст

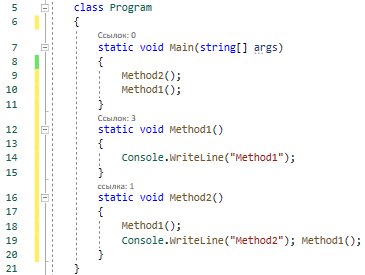
Автоматически созданное описание

Модификаторы и параметры необязательны.

Пример 1.

Определим два метода. Перед названием метода идет возвращаемый тип данных. Здесь это тип void, который указывает, что фактически ничего не возвращает, он просто производит некоторые действия.

После названия метода в скобках идет перечисление параметров. Но в данном случае скобки пустые, что означает, что метод не принимает никаких параметров.



В результате получим:



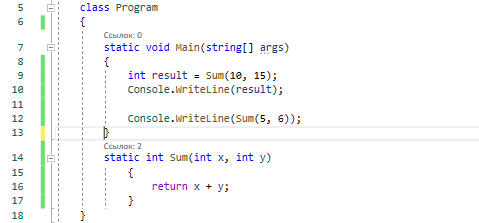
## Возвращение значения и оператор return

Оператор return останавливает выполнение метода и передаёт управление вызвавшему данный метод оператору. Если метод не void, то необходимо вернуть значение соответствующего типа.

Каждый метод имеет набор своих локальных переменных, они видны только в нем и при завершении работы метода уничтожаются. Для того чтобы переменные были видны в нескольких методах класса, необходимо объявить их полями вне метода, но внутри класса.

Для не void методов необходимо возвращать значение, каждый «путь выполнения» метода должен заканчиваться оператором return. Для void методов оператор return не обязателен.

Пример 2. Нахождение суммы двух чисел





## Использование параметров

Параметры позволяют передавать информацию из одного метода в другой. При объявлении метода можно задать список его параметров, если список пустой, то это означает, что метод не имеет параметров.

static void MethodWithParameters(int n, string y) {

// ... }

При вызове метода необходимо задать значения его параметрам. MethodWithParameters(2, "Hello, world");

или

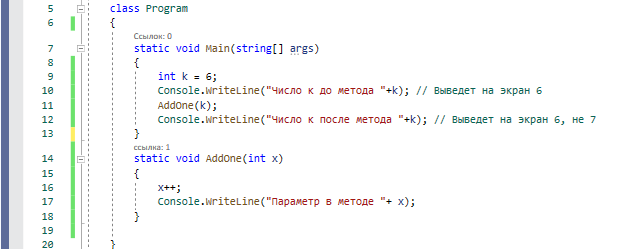
int p = 7;

string s = "Test message";

MethodWithParameters(p, s);

В С# существуют три варианта передачи параметров: по значению, по ссылке и выходные параметры. При передаче параметров по значению, изменение значения параметра в методе не влияет на значение в вызвавшем методе.

Пример 3

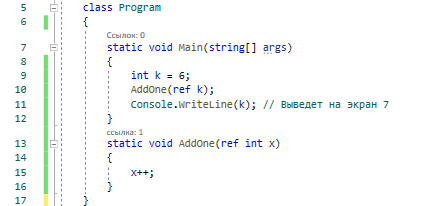


Изображение выглядит как текст, оранжевый, закрыть

Автоматически созданное описание

При передаче по ссылке передается ссылка на переменную, поэтому все действия, производимые с параметром, оказывают влияние на значение переменной в вызывающем методе. Для каждого параметра по ссылке необходимо указывать ключевое слово ref. До вызова метода необходимо обязательно инициализировать переменную.

Пример 4



Выходные параметры похожи на параметры по ссылке, единственное отличие их состоит в том, что их можно не инициализировать до вызова метода. При передаче выходного параметра перед ним указывается ключевое слово out.

Пример 5

Изображение выглядит как текст

Автоматически созданное описание

Причем, как и в случае с **ref** ключевое слово **out** используется как при определении метода, так и при его вызове.

С# позволяет использовать механизм передачи списка параметров изменяемой длины. Для этого используется ключевое слово params. Правило использования: допуст***и***м только один список параметров, который должен быть массивом конкретного типа и размещаться последним в общем списке параметров.

Пример 6

Изображение выглядит как текст

Автоматически созданное описание

Для выбора вида параметров необходимо учитывать два аспекта:

механизм передачи и эффективность. По эффективности передача по значению лучше, чем передачи по ссылке.

Когда метод вызывает себя, то это называется рекурсией. Если это происходит при участии другого метода, то это будет неявной рекурсией.

Рекурсия часто используется для упрощения логики программы.

## Порядок проведения работы

**Задание** 1. Ответить на контрольные вопросы в тетради:

1. Объясните что такое методы и почему они важны?
2. Назовите общее описание метода
3. Опишите три возможных пути передачи параметров и соответствующие ключевые слова С#.

**Задание 2.** Даны три числа, найти максимальное.Написать метод, возвращающий наибольшее из двух чисел. Входные параметры метода – два целых числа.

**Задание 3.** Написать метод, который меняет местами значения двух передаваемых параметров. Параметры передавать по ссылке.

**Задание 4.**

**Индивидуальные задания**

1. Даны 2 целочисленных числа а и Ь. Описать функцию, определяющую значение следующего выражения:

Y=max( a, 3b+4)\*max(3-2a, b-7).

1. Даны основания и высоты двух трапеций. Определить, у какой трапеции площадь больше.
2. Даны координаты вершин треугольника и радиус круга. Определить площади фигур.
3. Даны 3 числа. Найти наибольший общий делитель от этих чисел, используя созданную функцию определения наибольшего общего делителя от двух чисел.
4. Даны 2 числа. Найти наименьшее общее кратное.
5. Дан целочисленный массив. Найти сумму и произведение всех элементов массива, которые стоят на четных местах.
6. Даны 2 строки. Определить, какая строка имеет наибольшую длину. Не используя встроенную функцию Length класса String, описать функцию определения длины строки.
7. Составить процедуру, изображающую на экране Ваше имя с помощью символов «\*». Каждую букву оформить в виде отдельной процедуры.
8. Дан двумерный массив. Для каждой его строки определить минимальный и максимальные элементы, описав соответствующие подпрограммы.