

# Модуль 1, Практическое занятие 3

**Передача параметров в методы**  
**Условный оператор.**  
**Переключатель**  
**Циклы**

# Задача 1

**Написать метод, переводящий оценку в баллах десятибалльной шкалы в аттестационную (четырёх балльную) шкалу:**

- \* 1, 2, 3 балла – неудовлетворительно;**
- \* 4,5 – удовлетворительно;**
- \* 6,7 – хорошо;**
- \* 8, 9, 10 – отлично.**

**Используйте переключатель.**

**В основной программе в получайте от пользователя оценки (целые числа из диапазона 1..10 и выводите значение в четырёх балльной шкале.**

# Задача 2

Написать метод **Function()** с двумя логическими параметрами, вычисляющий и возвращающий значение логического выражения (конкретное выражение в условии должно быть явно задано).

Логическое выражение:

**!(p & q) & !(p | !q)**

В основной программе построить таблицу истинности логического выражения, заданного методом.

Модифицируйте предложенный на слайде 4 код метода **Main()** так, чтобы на экран выдавалась таблица из нулей и единиц.

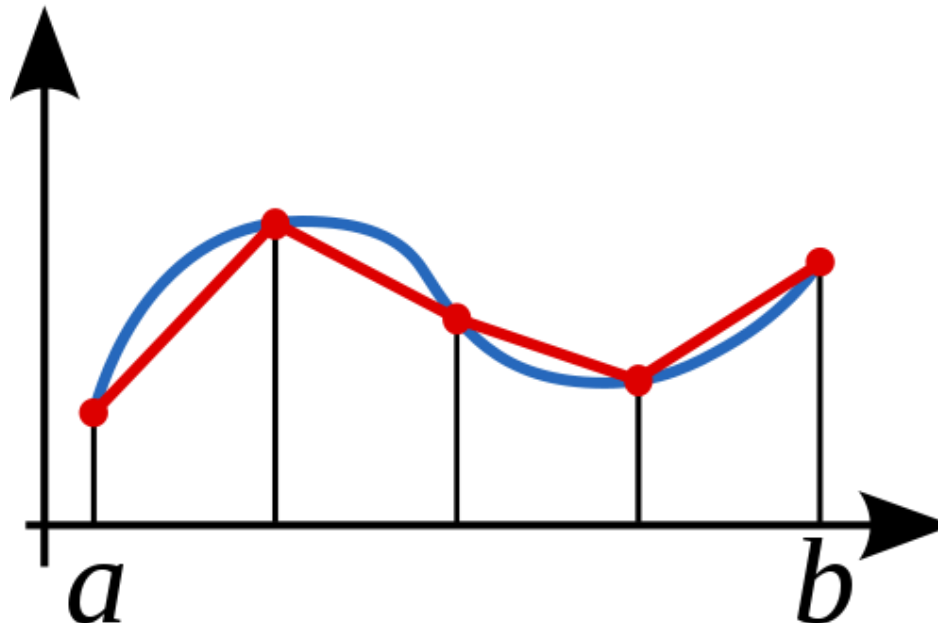
# Задача 2

```
using System;
class Program {
    static void Main() {
        bool p = true, q, res;
        Console.WriteLine("Таблица истинности !(p & q) & !(p | !q)");
        Console.WriteLine(" p \t q \t F");
        do {
            q = true;
            do {
                res = Function(p, q); // Вызов Function()
                Console.WriteLine("{0}\t{1}\t{2}", p, q, res);
                q = !q;
            } while (!q);
            p = !p;
        } while (!p);
        Console.WriteLine("Для выхода нажмите ENTER");
        Console.ReadLine();
    }
    // TODO: описание метода Function()
}
```

# Задача 3

Вычислить площадь под графиком функции  $x^2$  на отрезке  $[0;A]$  при помощи метода трапеций, вещественная точка  $A$  и шаг интегрирования  $\delta$  задаются с клавиатуры.

- Чтобы организовать проверку корректности введённых данных, определите ограничения на значения  $A$  и  $\delta$ .
- Как вычисляется значение, добавляемое к интегральной сумме на каждом шаге.
- Определите условие выхода из цикла формирования интегральной суммы.



# Задача 4

Написать метод для вычисления по формуле Ньютона с точностью до «машинного нуля» приближенного значения арифметического квадратного корня.

**Параметры:** подкоренное значение, полученное значение корня и значение точности, достигнутой при его вычислении. Если подкоренное значение отрицательно - метод должен возвращать в точку вызова значение **false**, иначе - **true**.

В основной программе вводить вещественные числа и выводить их корни. При отрицательных числах выводить сообщения.

Арифметическим корнем  $n$ -ной степени  $\sqrt[n]{A}$  положительного действительного числа  $A$  называется положительное действительное решение уравнения

$$x^n = A$$

(для целого  $n$  существует  $n$  комплексных решений данного уравнения, если  $A > 0$ , но только одно является положительным действительным).

Существует быстросходящийся **алгоритм нахождения корня  $n$ -ной степени**:

1. Сделать начальное предположение  $x_0$ ;

2. Задать 
$$x_{k+1} = \frac{1}{n} \left[ (n-1)x_k + \frac{A}{x_k^{n-1}} \right];$$

3. Повторять шаг 2, пока не будет достигнута необходимая точность.

[https://ru.wikipedia.org/wiki/Алгоритм\\_нахождения\\_корня\\_n-ной\\_степени](https://ru.wikipedia.org/wiki/Алгоритм_нахождения_корня_n-ной_степени)

# Задача 4

```
// Корень по формуле Ньютона
using System;
class Program {
    static void Main( ) {
        double x, result = 0, eps = 0;
        Console.Title = "Формула Ньютона";
        ConsoleKeyInfo клавиша; //Нажатая пользователем клавиша
        do {
            do {
                Console.Clear(); // очистка консольного окна
                Console.Write("x=");
            } while (!double.TryParse(Console.ReadLine(), out x));
            // TODO: вычисления (обращение к методу Newton) и вывод
            Console.WriteLine("Для выхода нажмите клавишу ESC");
            клавиша = Console.ReadKey(true);
        } while (клавиша.Key != ConsoleKey.Escape);
        Console.Beep(500, 1000);
    }
    // TODO: Объявление метода Newton()
}
```

# Задача 4

```
// ВЫЧИСЛЕНИЯ И ВЫВОД
```

```
if (!Newton(x, out result, out eps)) {  
    Console.WriteLine("Error!"); return;  
}
```

```
Console.WriteLine("root({0}) = {1,8:f4}, eps = {2,8:e4}", x, result, eps);
```

```
static bool Newton(double x, out double sq, out double eps)    {  
    double r1, r2 = x;  
    sq = eps = 0.0;  
    if (x <= 0.0) { Console.WriteLine("Ошибка в данных!");  
        return false; }  
    do    {  
        r1 = r2;  
        eps = x / r1 / 2 - r1 / 2;  
        r2 = r1 + eps;  
    } while (r1 != r2); // пока приближения «различимы» для ЭВМ  
    sq = r2;  
    return true;  
}
```



# Задача 5

Вычислите приближённое значение бесконечной суммы:

$$\frac{1}{1 \times 2 \times 3} + \frac{1}{2 \times 3 \times 4} + \frac{1}{3 \times 4 \times 5} + \dots$$

Протестируйте программу для типов данных **float** и **double**. В чём различие результатов?

# Задача 6

Написать метод для расчета сложных процентов.

**Параметры:** начальный капитал, годовая процентная ставка, число лет (вклада).

**Возвращаемое значение** – итоговая сумма в конце срока вклада.

**static double Total(double k, double r, uint n)**

В основной программе ввести начальный капитал (большой нуля), процентную ставку и число лет. Вывести таблицу значений итоговых сумм в конце каждого года вплоть до заданного числа лет.

# Задача 6

```
using System;
class Program {
    static void Main() {
        double k, r, s, temp;
        uint n;
        do Console.Write("Введите начальный капитал: ");
        while (!double.TryParse(Console.ReadLine(), out k)
            | k <= 0); // Капитал не отрицателен
        do Console.Write("Введите годовую процентную ставку: ");
        while (!double.TryParse(Console.ReadLine(), out r)
            | r <= 0); // Процент не отрицателен
        do Console.Write("Введите число лет: ");
        while (!uint.TryParse(Console.ReadLine(), out n)
            | n == 0); // число лет не равно нулю

        s = Total(k, r, n); // обращение к методу
        Console.WriteLine("Итоговая сумма: " + s);
    } // end of Main()
} // end of Program
```

# Задача 7 (самостоятельно)

Написать метод для решения квадратного уравнения.

Параметры – коэффициенты уравнения **A**, **B**, **C**, и два параметра, для получения значений вещественных корней.

При отсутствии вещественных корней (если **A=B=C=0** или **A=B=0** и **C!=0**) метод должен возвращать в точку вызова значение **false**, иначе - **true**.

В основной программе вводить коэффициенты квадратного уравнения, выводить значения вещественных корней или сообщение об их отсутствии.

# Выполните самостоятельно дома или на семинаре

1) Написать метод, находящий трехзначное десятичное число **s**, все цифры которого одинаковы и которое представляет собой сумму первых членов натурального ряда, то есть  **$s = 1+2+3+4+\dots$**

Вывести полученное число, количество членов ряда и условное изображение соответствующей суммы, в которой указаны первые три и последние три члена, а средние члены обозначены многоточием.

*Например, если последний член равен 25, то вывести:*

*$1+2+3+\dots+23+24+25$ .*

2) Написать метод, преобразующий число переданное в качестве параметра в число, записанное теми же цифрами, но идущими в обратном порядке.

*Например, 1024 -> 4201, 120 -> 21*

2) Модифицируйте коды заданий из семинара, включив в них проверки корректности и осмысленности введенных данных.

# Выполните самостоятельно дома или на семинаре

1. Написать метод, вычисляющий логическое значение функции **G=F(X,Y)**.  
Результат равен **true**, если точка с координатами (X,Y) попадает в фигуру G, и результат равен **false**, если точка с координатами (X,Y) не попадает в фигуру G. Фигура G - сектор круга радиусом R=2 в диапазоне углов  $-90 \leq \varphi \leq 45$ .

2. Написать метод, вычисляющий значение функции  $G=F(X,Y)$

$$G = \begin{cases} X + \sin(Y), & X < Y \text{ и } X > 0 \\ Y - \cos(X), & X > Y \text{ и } X < 0 \\ 0.5 \cdot X \cdot Y, & \text{в остальных случаях} \end{cases}$$

1. Написать метод, вычисляющий значение функции  $G=F(X)$

$$G = \begin{cases} \sin\left(\frac{\pi}{2}\right), & X \leq 0.5 \\ \sin\left(\frac{\pi \cdot (x - 1)}{2}\right), & X > 0.5 \end{cases}$$

1. (\*\*\*) Трехзначным целым числом кодируется номер аудитории в учебном корпусе. Старшая цифра обозначают номер этажа, а две младшие – номер аудитории на этаже. Из трех аудиторий определить и вывести на экран ту аудиторию, которая имеет минимальный номер внутри этажа. Если таких аудиторий несколько - вывести любую из них.