Материалы для подготовки к семинару 3, модуль 1

# Условный оператор

## Полная форма

<условие>

<оператор\_1>

<оператор\_2>

true

false

**if (x > 0) {**

**x = x;**

**}**

**else {**

**x = -x;**

**}**

## Неполная форма

<условие>

<оператор\_1>

true

false

**if (true) {**

**Console.Write(“11”);**

**}**

## Примеры

### Пример 1.

Ввести значение вещественной переменной ***х*** и вычислить (и вывести) значение функции, заданной соотношениями:

using System;

class Program {

static void Main() {

double x, y;

do { // цикл для повторения решений задачи

// Конструкция для ввода значения:

do Console.Write("Введите аргумент функции: ");

while (!double.TryParse(Console.ReadLine(), out x));

if (x < 0) y = 1 / x / x;

else

if (x == 0) y = 12;

else

if (x < 10) y = x \* x;

else y = 100;

Console.WriteLine("F(" + x + ") = " + y);

Console.WriteLine("Для выхода нажмите клавишу ESC");

} while (Console.ReadKey(true).Key != ConsoleKey.Escape);

}

}

# Методы

Повторяющийся (единообразный код) удобно оформлять в виде подпрограммы (метода). Методы в этом семинаре должны быть объявлены в классе как статические.

**class Program {**

**static void Main() {**

**double a = 1.44,**

**b = 0.25;**

**double summDbl = 2 \* a + 2 \* b;**

**System.Console.Write("a = {0} b = {1} 2\*a+2\*b= {2}", a, b,summDbl);**

**}**

**}**

Выделим вычисления выражения **2 \* a + 2 \* b** в отдельный метод с двумя вещественными параметрами. В правой части операции присваивания **double summDbl = 2 \* a + 2 \* b;** вычисление выражения заменим вызовом метода.

**class Program {**

**// новый метод может быть описан до или после**

**// своего первого вызова**

**static double Summ(double a, double b) {**

**// a, b - локальны для Summ()**

**return 2 \* a + 2 \* b; // возращаем значение в точку вызова**

**}**

**static void Main() {**

**double a = 1.44,**

**b = 0.25;**

**double summDbl = Summ(a, b); // вызов Summ()**

**System.Console.Write("a = {0} b = {1} 2\*a+2\*b= {2}", a, b,summDbl);**

**}**

**}**

**class Program {**

**static double Summ(double a, double b) {**

**a = 2 \* a;**

**b = 2 \* b;**

**return a + b;**

**}**

**static void Main() {**

**double a = 1.44,**

**b = 0.25;**

**double summDbl = Summ(a, b);**

**System.Console.Write("a = {0} b = {1} summDbl = {2}",**

**a, b, summDbl);**

**}**

**}**

параметры

аргументы

# Возвращение значения в точку вызова

Для возвращения значения в точку вызова метода в его теле используется оператор **return**. Если метод не возвращает значения, то **return** может быть опущен, а тип возвращаемого значения указывается как **void**.

**…**

**double a = 1.44,**

**b = 0.25;**

**double summDbl = Summ(a, b);**

**System.Console.Write("a = {0} b = {1} summDbl = {2}", a, b, summDbl);**

**…**

**static double Summ(double a, double b) {**

**a = 2 \* a;**

**b = 2 \* b;**

**return a + b;**

**}**

Поток выполнения программы

Создаются копии значений, хранящихся в переменных (параметрах) **a** и **q**.

Управление передаётся методу **Summ()**.

Вычисленное в методе значение передается в точку его вызова.

*sum(а = 1,44, b = 0,25)*

*{*

*а = 2\*1,44 = 2,88*

*b = 2\*0,25 = 0,5*

*return 2,88+0,5 = 3,38*

*}*

**В точку вызова**

### Задание к примеру 1

Выделить из программы метод, вычисляющий значение определенной в условии математической функции. Заголовок метода: **double Function(double x)**. Основная программа, используя метод, «общается» с пользователем.

### Пример 2. Переключатель (множественный выбор)

Написать метод, выполняющий арифметические операции над значениями двух параметров. Третий параметр – символ операции.

В основной программе, вводить значения двух переменных целого типа и символ операции, обращаясь к методу, вычислять значение соответствующего выражения.

using System;

class Program {

static void Main() {

int firstOperand, secondOperand;

do {

// получаем значения операндов

do Console.Write("Введите целое число: ");

while (!int.TryParse(Console.ReadLine(), out firstOperand));

do Console.Write("Введите целое число: ");

while (!int.TryParse(Console.ReadLine(), out secondOperand));

// получаем знак операции

char operationSign = ' ';

do Console.Write("Введите знак операции: ");

while (!char.TryParse(Console.ReadLine(), out operationSign)

|| (operationSign != '+'

& operationSign != '-'

& operationSign != '\*'

& operationSign != '/'));

int result = Expression(firstOperand, secondOperand, operationSign);

Console.WriteLine("result = " + result);

Console.WriteLine("Для выхода нажмите ESC.....");

} while (Console.ReadKey(true).Key != ConsoleKey.Escape);

}

// Метода Expression()

// Выполняет бинарные арифметические операции

// над целыми числами

static int Expression(int op1, int op2, char operation) {

int res = 0;

switch (operation) {

case '+': res = op1 + op2; break;

case '-': res = op1 - op2; break;

case '\*': res = op1 \* op2; break;

case '/': res = op1 / op2; break;

default: Console.WriteLine("Error!"); break;

}

return res;

}

}

### Задание к примеру 2

Дополнить программу операцией получения остатка от деления двух чисел, ОБОЗНАЧАЯ ЕЕ СИМВОЛОМ ‘%’. Для вычисления остатка от деления написать метод, встроенную операцию получения остатка от деления не использовать.

# Циклы

## Цикл с предусловием

**<выражение\_условие>**

**<тело\_цикла>**

false

true

**int i = 5;**

**while (i > 1) {**

**i--;**

**}**

## Цикл с постусловием

**<выражение\_условие>**

**<тело\_цикла>**

false

true

**int i = 5;**

**do {**

**i--;**

**} while (i > 1);**

## Примеры

### Пример 1

// Ввести натуральное число и напечатать его цифры "столбиком".

using System;

class Program {

static void Main() {

uint numberFromUser, // Введенное число

currFigure, // Очередная цифра

div = 1, // Делитель

temp; // рабочая переменная

do {// цикл для повторения решений задачи

do

Console.Write("Введите целое число больше нуля: ");

while (!uint.TryParse(Console.ReadLine(), out numberFromUser)

| numberFromUser == 0); // Ноль или меньше!

while (numberFromUser >= div) div \*= 10; // цикл с предусловием

// div - наименьшая степень 10-ти, не меньшая numb.

temp = numberFromUser;

do // цикл с постусловием

{

div /= 10;

currFigure = temp / div; // очередная цифра числа

Console.WriteLine(currFigure);

temp = temp - currFigure \* div; // удаляем старшую цифру

} while (div > 1);

Console.WriteLine("Для выхода нажмите клавишу ESC");

} while (Console.ReadKey(true).Key != ConsoleKey.Escape);

}

}

### Задание к примеру 1

1. Для чего в коде программы использована переменная div?
2. Измените программу так, чтобы поставленная задача достигалась без использования переменной div.

### Пример 2

Сократить введенную обыкновенную дробь. Дробь вводится с клавиатуры в виде целых положительных значений числителя и знаменателя.

using System;

class Program {

static void Main() {

uint numerator, //Числитель исходной дроби

denominator, //знаменатель исходной дроби

GCD, // НОД

reducedNumerator, // сокращённый числитель

reducedDenominator; // сокращённый знаменатель

do {

do Console.Write("Введите числитель: ");

while (!uint.TryParse(Console.ReadLine(), out numerator)

|| numerator == 0); // Должно быть положительное число

do Console.Write("Введите знаменатель: ");

while (!uint.TryParse(Console.ReadLine(), out denominator)

|| denominator == 0); // Должно быть положительное

uint x = numerator, y = denominator;// Вспомогательные переменные

while (x != y) // Вычисляем НОД

if (x > y) x -= y;

else y -= x;

GCD = x;

reducedNumerator = numerator / GCD;

reducedDenominator = denominator / GCD;

Console.WriteLine(numerator + "/" + denominator + " ===> " +

reducedNumerator + "/" + reducedDenominator);

Console.WriteLine("Для выхода нажмите клавишу ESC");

} while (Console.ReadKey(true).Key != ConsoleKey.Escape);

}

}

### Задание к примеру 2

Нахождение НОД двух целых положительных чисел выделите в отдельный метод GCD(), параметры и тип возвращаемого значения метода определите самостоятельно. В коде примера 2 используйте вызов метода GCD().

## Универсальный цикл (for)

## Примеры

<инициализатор\_цикла>

<выражение\_условие>

<тело\_цикла>

<завершающее\_выражение>

true

false

**int i = 3;**

**for (int j = 0; j < 7; j++) {**

**i = j \* 2;**

**}**

### Пример 1

Ввести натуральное число и вычислить суммы его цифр, находящихся на четных и на нечетных позициях.

using System;

class Program {

static void Main() {

uint numb, // Введенное число

curFig, // Очередная цифра

sOdd, // сумма нечетных

sEven, // сумма четных

temp; // рабочая переменная

do {

do Console.Write("Введите натуральное число: ");

while (!uint.TryParse(Console.ReadLine(), out numb)

| numb == 0); // Нуль не включаем

sOdd = 0; // При повторении решения

sEven = 0; // нужны новые суммы

temp = numb; // не будем "портить" исходные данные

for (int k = 0; temp >= 1; k++, temp /= 10) {

curFig = temp % 10;

if (k % 2 == 0) // анализируем четность/нечетность позиции

sOdd += curFig;

else

sEven += curFig;

}

Console.WriteLine("Для числа {2} Суммы: четные позиции={1}

\tнечетные позиции={0}", sOdd, sEven, numb);

Console.WriteLine("Для выхода нажмите клавишу ESC");

} while (Console.ReadKey(true).Key != ConsoleKey.Escape);

}

}

### Задание к примеру 1

Выделить из программы методы вычисления сумм цифр, стоящих на чётных и нечётных позициях натурального числа - параметра. Минимизировать количество итераций цикла, необходимых для получения цифр.

# Задачи

## Задача 1

Написать метод вычисления площади треугольника по формуле Герона. Параметры – длины сторон и вычисляемая площадь. Если значения параметров не могут быть длинами сторон треугольника, метод должен возвращать в точку вызова значение **false**, иначе - **true**.

## Задача 2

Написать метод определения принадлежности точки заданной области на плоскости. Область задана соотношениями: -10 < **y** < 10 и **y** < **-x** +10 и **y** > **-x** – 10. В основной программе, вводя значения координат, проверять принадлежность точки области.

## Задача 3

Задан круг с центром (10; 10) и радиусом 5. Ввести координаты центра и радиус второго круга. Если круги пересекаются, ввести координаты новой точки и оценить ее принадлежность пересечению кругов. Проверку пересечения кругов выполнять в методе **IsCrossed()**, оценку принадлежности точки пересечению кругов в методе **IsLaidIn()**.

## Задача 4

Куб с известным ребром из вещества с заданной плотностью погружен в воду (гранью вниз). Напишите метод, выводящий в консольное приложение сообщение «Тонет» или «Плавает». Если плавает – вычислите «осадку» (расстояние от дна куба до поверхности воды).

## Задача 5

Написать метод, вычисляющий значение функции ***Y=F(X)***, заданной требованиями:

## Задача 6

Вывести таблицу значений функции двух переменных ***z = F(x,y)***. Для этого описать метод **TabPrint()** без параметров, возвращающий значение типа void. Из основной программы вызывать **TabPrint()**.