

Тема: Типы данных и переменные

Основы программирования

Составил: Карышев А.А.

Типы данных

Ключевое слово языка C#	Тип .Net	Вид значения	Используемая память	Диапазон
sbyte	SByte	Целое число	8	-128...127
byte	Byte	Целое число	8	0...255
short	Int16	Целое число	16	-32768...32767
ushort	UInt16	Целое число	16	0...65535
int	Int32	Целое число	32	$-2^{31} \dots 2^{31}$
uint	UInt32	Целое число	32	$0 \dots 2^{32}$
long	Int64	Целое число	64	$-2^{63} \dots 2^{63}$
ulong	UInt64	Целое число	64	$0 \dots 2^{64}$
char	Char	1 символ	16	Все символы Unicode
float	Single	Числа с плавающей точкой	32	$(+/-)1.5 \cdot 10^{-45} \dots (+/-) 3.4 \cdot 10^{38}$, 7 разрядов
double	Double	Числа с плавающей точкой	64	$(+/-)5 \cdot 10^{-324} \dots (+/-) 3.4 \cdot 10^{30}$, 16 разрядов
decimal	Decimal	Числа с плавающей точкой повышенной точности	128	$(+/-)1 \cdot 10^{-28} \dots (+/-) 7.9 \cdot 10^{28}$, 30 разрядов
bool	Boolean	Логический	1	false, true

Переменные

Переменная — это расположение в памяти объекта определенного типа. Переменные могут иметь значения, которыми они проинициализированы, или эти значения могут быть изменены программно.

Для объявления переменной используется следующий синтаксический блок:

```
<объявление_переменной>::=  
<тип> <идентификатор1>,[<идентификатор2>]...
```

Для присваивания переменной значения введенного с клавиатуры используется следующий синтаксический блок:

```
<ввод_чисел_с_клавиатуры>::= <идентификатор_переменной>  
=<Тип_.Net_CTS>.Parse(Console.ReadLine());
```

Если объявление переменной идёт непосредственно перед присвоением ей начального значения, эти две операции можно объединить:

```
<объяв_переменной_с_присвоением_начального_значения>::=  
<тип> <идентификатор_переменной> = <начальное_значение>
```

Бинарные операции

Бинарная операция – это операция, производимая над двумя операндами.

$\langle \text{операнд} \rangle ::= \langle \text{литерал} \rangle$

$::= \langle \text{идентификатор_переменной} \rangle$

$::= \langle \text{числовое_выражение} \rangle$

$::= \langle \text{вызов_метода} \rangle$

$\langle \text{вызов_метода} \rangle$ - обращение к заранее написанному фрагменту кода для нахождения некоторого значения или осуществления некоторого действия.

$\langle \text{числовое_выражение} \rangle ::= \langle \text{операнд} \rangle \langle \text{бинарная_операция} \rangle \langle \text{операнд} \rangle$

Арифметические операции

Приоритет	Назначение	Обозначение	Название
1	Установка порядка действий	(<числовое_выражение>)	круглые скобки
	Инкремент	<идентификатор_переменной> ++	инкремент в постфиксной форме
	Декремент	<идентификатор_переменной> --	декремент в постфиксной форме
2	Ничего	+ <операнд>	Унарный плюс
	Изменение знака	- <операнд>	Унарный минус
	Инкремент	++ <идентификатор_переменной>	Инкремент в префиксной форме
	Декремент	-- <идентификатор_переменной>	Декремент в префиксной форме
3	Умножение	<операнд> * <операнд>	Умножение
	Деление	<операнд> / <операнд>	Если операнды – целые числа, то деление – целочисленное, если дробные – то просто деление.
	Деление по модулю	<операнд> % <операнд>	Получение остатка от деления
4	Сложение	<операнд> + <операнд>	Сложение
	Вычитание	<операнд> - <операнд>	Вычитание

Операции сравнения

Синтаксис C#	Название	Пример	Категория	Объяснение
>	Больше	$x > y$	Операция отношения	Возвращает true, если x больше y, иначе — false
>=	Больше или равно	$x >= y$	Операция отношения	Возвращает true, если x больше или равно y, иначе — false
<	Меньше	$x < y$	Операция отношения	Возвращает true, если x меньше y, иначе — false
<=	Меньше или равно	$x <= y$	Операция отношения	Возвращает true, если x меньше или равно y, иначе — false
==	Равно	$x == y$	Операция равенства	Возвращает true, если x равно y, иначе — false
!=	Не равно	$x != y$	Операция равенства	Возвращает true, если x не равно y, иначе — false

Выражение отношения

Операции отношения формируют выражения отношения, а операции равенства, соответственно, выражения равенства.

$\langle \text{Выражение_отношения} \rangle ::=$

$\langle \text{Операнд} \rangle \langle \text{Операция_отношения} \rangle \langle \text{Операнд} \rangle$

$\langle \text{Выражение_равенства} \rangle ::=$

$\langle \text{Операнд} \rangle \langle \text{Операция_равенства} \rangle \langle \text{Операнд} \rangle$

$\langle \text{Операнд} \rangle$

$::= \langle \text{Литерал} \rangle$

$::= \langle \text{Идентификатор_числовой_переменной} \rangle$

$::= \langle \text{Идентификатор_числовой_константы} \rangle$

$::= \langle \text{Числовое_выражение} \rangle$

$::= \langle \text{Вызов_метода} \rangle$

Алгоритмы

Алгоритм – это некоторая последовательность действий, которая за конечное число шагов позволяет получить результат или прийти к выводу, что получение результата невозможно.

начало

блок начала алгоритма (должен быть только один)

Конец

блок окончания выполнения алгоритма
(должно быть не менее одного)

I/O

блок ввода/вывода

< Действие

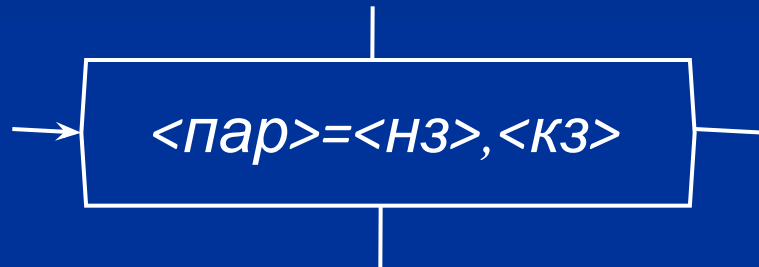
Блок действия. Любые вычисления
оказываются в таких блоках.

>

Синтаксический блок



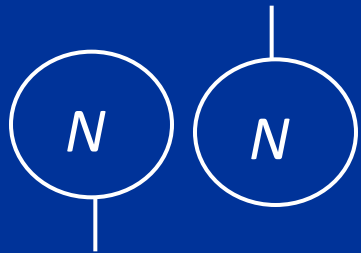
блок условия (условного перехода)



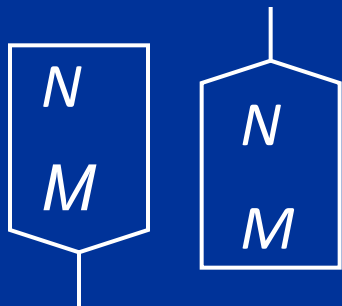
цикл с параметрами

<НЗ> - начальное значение

<КЗ> - конечное значение



внутристраничный переход N – номер перехода



Межстраничный переход

N – номер страницы, на которую осуществляется переход;

M – номер перехода на странице.

Пример алгоритма.

Задание. Даны два числа вводимых с клавиатуры.

Вычислить и вывести на экран: сумму, разность, произведение и частное.

