

Тема: Алфавит С#. Типы данных.

Основы программирования

Составил: Карышев А.А.

Алфавит C#

Алфавит C# включает в себя:

1. Строчные и прописные буквы латинского алфавита.
2. Цифры от 0 до 9.
3. Символ `_` (подчеркивание — также считается буквой);
4. Набор специальных символов: `" { }, | [] + - % / \ ; ' :
? < > = ! & # *`.
5. Прочие символы.

Лексемы

Алфавит C# служит для построения слов, которые в C# называются лексемами.

Различают пять типов лексем:

1. Идентификаторы;
2. Ключевые слова;
3. Знаки (символы) операций;
4. Литералы;
5. Разделители.

Лексемы отделяются друг от друга специальными символами - **разделителями**. Этой же цели служит множество пробельных символов, к числу которых относятся пробел, табуляция, символ новой строки и комментарии.

Идентификаторы

Имена считают идентификаторами. Имена служат для обозначения переменных, типов данных, функций.

Рассмотрим правила построения идентификаторов состоящих из букв алфавита:

- первым символом идентификатора C# может быть только буква;
- следующими символами идентификатора могут быть буквы, цифры и нижнее подчеркивание;
- длина идентификатора не ограничена.

Ключевые слова

Часть идентификаторов $S\#$ входит в фиксированный словарь ключевых слов. Эти идентификаторы образуют подмножество ключевых слов (они так и называются ключевыми словами).

Прочие идентификаторы после специального объявления становятся именами.

Таким образом, любая случайным образом составленная последовательность букв, цифр и знаков подчеркивания с точки зрения грамматики языка идеально подходит на роль имени любого объекта, если только начинается с буквы

Типы данных C#

C# является жестко типизированным языком. При его использовании необходимо объявлять тип каждого объекта, который создаете (например, целые числа, числа с плавающей точкой, строки, окна, кнопки и т. д.), а компилятор поможет избежать ошибок, связанных с присвоением переменным значений только того типа, который им соответствует.

Проверка типов данных

При компиляции программы проходит обязательная проверка соответствия типов.

Соответствие типов проверяется в трёх случаях:

1. **Присваивание.** Например, чтобы нельзя было в целочисленную переменную записать дробную.

2. **При вычислении выражений** компилятор автоматически приводит тип результата выражения к типу, требующему наибольшее количество памяти аргументов выражения. Например, $2+3*1.5=6.5$

3. **При вызове методов.**

Разновидности типов данных

1. Язык программирования C# подразделяет типы на два вида: встроенные типы, которые определены в языке, и определяемые пользователем типы, которые выбирает программист.

2. C# также подразделяет типы на две другие категории: размерные и ссылочные. Основное различие между ними — это способ, с помощью которого их значения сохраняются в памяти.

Размерные типы сохраняют свое фактическое значение в стек. Ссылочные типы хранят в стеке лишь адрес объекта, а сам объект сохраняется в куче.

Простые типы.

Простыми типами называются «примитивные» типы, принадлежащие к группе встроенных предопределённых типов C#.

Каждый из типов характеризуется следующими признаками:

1. Какие значения можно хранить в переменных данного типа.
2. Диапазон значений.
3. Объём используемой памяти (для представления одной переменной требуется от 8 до 128 бит).
4. Типы операций, которые можно делать над переменными.

Целочисленные типы

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	$0 \dots 2^{64}$

Числа с плавающей точкой, СИМВОЛЬНЫЙ И ЛОГИЧЕСКИЙ ТИПЫ

Ключевое слово языка C#	Тип .Net	Вид значения	Используема я память	Диапазон
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

Совместимость типов

Часто несколько разных типов участвуют в одном и том же выражении, что может привести к ошибкам, связанным с несовместимостью типов.

Некорректные присваивания могут принадлежать к одному из двух видов:

1. Исходная величина превышает верхний предел для типа назначения. Это называется переполнением сверху (*overflow*).

2. Исходная величина меньше, чем нижний предел для типа назначения. Это называется переполнением снизу (*underflow*).