

[О типах](#)

[Различные типы](#)

[Разновидности типов](#)

[Особенности сред разработки](#)

О типах

Прежде, чем говорить непосредственно о программировании на C++, т.е. о том, как заставить компьютер делать "что-то" с "чем-то", давайте посмотрим, с чем же именно он может делать это самое "что-то".

Любая сущность, с которой может работать программа на C++, должна принадлежать к какому-то типу данных, который определяет, какие операции можно выполнять над этой сущностью и как эти операции интерпретируются. Можно сказать, что тип данных определяется операциями, которые можно над ним производить; также и различные операции могут производиться над конкретными типами определенным образом. Например, целые числа можно складывать, умножать, сравнивать. Брать синус можно от целых или дробных чисел, а от строк нельзя. Функции вывода на экран работают с числами и символами по-разному. Поэтому для того, чтобы программа могла правильно работать с вводимыми нами сущностями, они должны принадлежать к какому-то типу, который и определит правила работы с этой сущностью.

Различные типы

В C++ (согласно Стандарту C++), имеются следующие типы:

- логический тип (bool) - означает "истина" ("true") или "ложь" ("false"), используется в условных выражениях.
- символьные типы (например, char) - символы, такие как буквы, цифры и некоторые специальные символы.
- целочисленные типы (например, int) - целые положительные, отрицательные числа и ноль.
- типы с плавающей точкой (например, double) - то есть дробные числа.
- тип void, означающий отсутствие каких-либо данных (например, в Pascal'e функции, результат выполнения которых есть величина какого-либо типа, называются также "функциями", а функции, просто выполняющие некоторые инструкции и не возвращающие никакие величины - "процедурами". Функция типа void в C++ - тоже самое, что процедура в Pascal'e)

Можно также задавать перечисляемые типы данных (enum) - указывается набор всех возможных значений для величин этого типа. Например, для типа, означающего дни недели, можно задать значения Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday.

На основе указанных выше типов можно конструировать и другие:

- массивы элементов одного типа (например, двумерный массив целых чисел `int[][]`)
- указатели на величины определенного типа (например, указатель на символ `char*`)
- ссылки на величины не менее определенного типа (например, ссылка на число с плавающей точкой `double&`)

Также по своему произволу программист может определять свои собственные типы - в виде структур (`struct`) и классов (`class`), которые есть во многом схожие друг с другом понятия.

В стандартной библиотеке имеется специальный тип `string` для строк, являющийся заменой массиву символов `char[]`, использовавшемуся для работы со строками в C. По сути он является классом, и обеспечивает многие полезные функции.

Разновидности типов

Не все типы одинаково полезны. Для подавляющего большинства случаев если вам нужно использовать символьный тип, используйте `char`, если нужен целочисленный тип - `int`, если тип с плавающей точкой - `double`. Все прочие типы этих категорий могут понадобиться только в специальных случаях, поэтому как правило о них нужно просто знать что они есть, чтобы читать чужие программы или если они действительно окажутся в какой-то ситуации полезны.

К символьным типам относятся `char`, `signed char` и `unsigned char`. Отличие `signed` от `unsigned` в том, что при преобразовании к целому типу первый будет гарантированно знаковым (например, в диапазоне от -128 до +127), а второй гарантированно беззнаковым (например от 0 до +255).

Целочисленные типы, помимо `int`, бывают, с одной стороны `signed int` (синоним - `signed`) и `unsigned int` (синоним - `unsigned`), т.е. знаковый и беззнаковый, с другой стороны - `short int` (синоним - `short`) и `long int` (синоним - `long`), т.е. имеющие различные размеры. Тип `int` является знаковым.

Типы с плавающей точкой бывают трех размеров - `float` (одинарной точности), `double` (двойной точности) и `long double` (расширенной точности).

Конкретные размеры различных типов зависят от реализации и архитектуры компьютера.

Особенности сред разработки

В Borland C++ Builder для строк имеется также тип `AnsiString`, во многом сходный с типом `string`.

В Borland C++ 3.1, во-первых, не существует типа `string` из стандартной библиотеки, поэтому для хранения строк используется массив символов. Во-вторых, в нем нет типа `bool`, поэтому для хранения логических переменных используется `int`, что в целом менее наглядно и прозрачно. Вместо "false" выступает значение "0", вместо "true" - как правило отрицание нуля - "!0". Соответственно, в логических выражениях происходит сравнение на равенство нулю квазилогических переменных и выражений из них.

Урок 1 - Обзор типов данных

Автор: Андрей

04.04.2009 16:05 - Обновлено 27.10.2009 16:02
