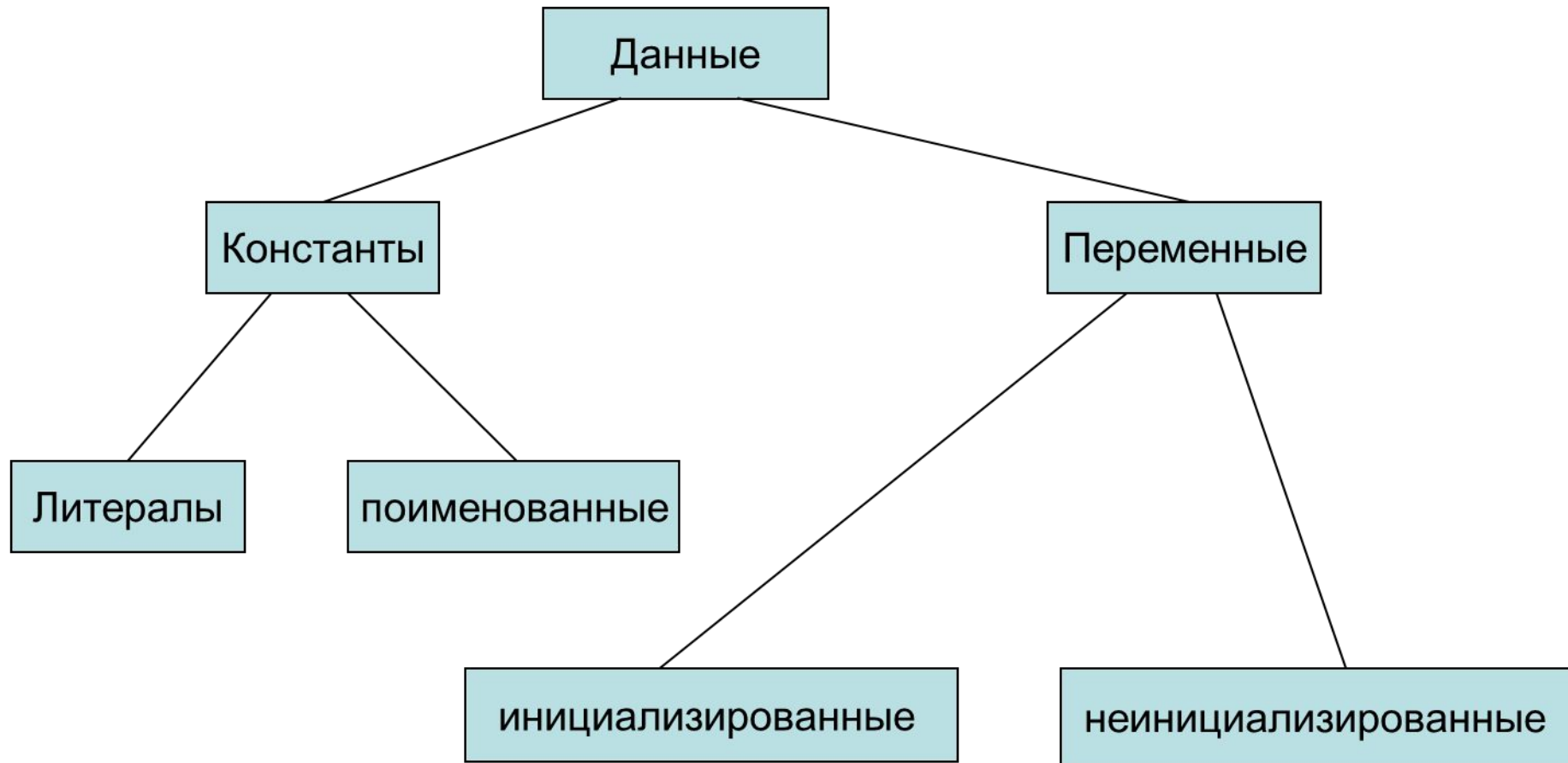


ТИПЫ ДАННЫХ КОНСТАНТЫ C++

Основными объектами любой программы являются данные



Константы

Константы – данные, не изменяемые в процессе выполнения программы.

Поименованные константы – константы, обращение к которым выполняется по имени. Они описываются в разделе описаний.

Литералы – это лексема, представляющая изображение фиксированного числового, строкового или символьного значения, записанная в тексте программы.

Константы делятся на пять групп:

- целые,
- вещественные,
- перечислимые,
- символьные,
- строковые.

Компилятор, выделив константу, относит ее к той или другой группе по ее «внешнему виду» (по форме записи) в исходном тексте и по числовому значению.

Константы

Целые константы могут быть десятичными, восьмиричными и шестнадцатиричными.

Десятичная константа определена как последовательность десятичных цифр, начинающаяся не с нуля, если это число не нуль. Может быть отрицательной и положительной.

Пример: 16, 56783, 0, -567, 7865.

Восьмиричная константа определена как последовательность десятичных цифр от 0 до 7, всегда начинающаяся с нуля. Может быть отрицательной и положительной.

Пример: 016, 020, 0777,

Шестнадцатиричная константа определена как последовательность шестнадцатиричных цифр, которая начинается сочетанием 0x. Может быть отрицательной и положительной.

Пример: 0x30, 0xF, 0xe, 0x56AD.

В зависимости от значения целой константы компилятор представляет ее в памяти в соответствии с типом. Для явного указания способа представления программист может использовать суффиксы L, l или U, u (64L, 0b7u, 0x56L).

Константы

Вещественные константы представлены в формате с плавающей точкой.

Константа с плавающей точкой может включать семь частей:

- целая часть (десятичная целая константа);
- десятичная точка ;
- дробная часть (десятичная целая константа) ;
- признак экспоненты (символ e или E);
- показатель десятичной степени (десятичная целая константа, возможно со знаком) ;
- суффикс F(или f) либо L(или l).

В записи вещественного числа могут опускаться целая или дробная часть (но не одновременно), десятичная точка или признак экспоненты с показателем степени, суффикс.

Пример: 66. .045 .0 3.1459F 1.34e-12 45E+6L 56.891

Без суффиксов F или L под вещественную константу отводится 8 байт.

Константы

Символьные константы – это один или два символа, заключенные в апострофы.

Примеры:

'Z' '*' '\$' '\012' '\0' '\n' – односимвольные константы.

'db' '\x07\x07' '\n\t' - двухсимвольные константы.

Символ '\' используется для:

- записи кодов, не имеющих графического изображения
- символов ('),(\),(?),(")
- задания символьных констант, указывая их коды в 8-ричном или 16-ричном виде.

Последовательность символов, начинающаяся с символа '\' называется эскейп-последовательностью.

Константы

Строка или строковая константа определяется как последовательность символов, заключенная в кавычки.

Пример:

“Это пример строки, называемой строковой константой”

Среди символов строки могут быть эскейп-последовательности, то есть сочетания, соответствующие неизображаемым символьным константам или символам, задаваемых их внутренними кодами. В этом случае они начинаются с символа `\`.

`“\nЭто строка,\nпначе -\”строинг\”,\nпначе - \”строковый литерал\”.”`

Перечислимые константы по существу (по внутреннему представлению) являются обычными целыми константами, которым приписаны уникальные и удобные для использования обозначения. Будут рассмотрены далее.

ПЕРЕМЕННЫЕ

Переменные

Переменные – поименованные данные, которые могут изменяться в процессе выполнения программы.

Переменные характеризуются именем и значением.

Именем служит идентификатор.

Переменная – это частный случай объекта как поименованной области памяти. Отличительной чертой переменной является возможность связывать с ее именем различные значения, совокупность которых определяется типом переменной.

При определении значения переменной в соответствующую ей область памяти помещается некоторый код.

Это может происходить:

- во время компиляции, тогда переменная называется инициализированной (`int s=56`);
- во время выполнения программы, тогда переменная называется неинициализированной (`char C`).

Переменная типизируется с помощью определений и описаний.

Типы данных

Тип – описатель данных, который определяет:

- а) диапазон изменения значения, задавая размер ее внутреннего представления;
- б) множество операций, которые могут выполняться над этой переменной
- в) требуемое для переменной количество памяти при ее начальном распределении
- г) интерпретацию двоичного кода значений при последующих обращениях к переменным.

Кроме того, тип используется для контроля типов с целью обнаружения возможных случаев недопустимого присваивания.

В C++ стандартно определено большое количество типов, которые программист может использовать без предварительного описания.

Фундаментальные типы данных.

Интегральные типы

Имя типа	Подтипы	Размер , байт	Интервал значений
char или _int[8]	[signed] char unsigned char	1	-128..127 0..255
short или _int[16]	[signed] short unsigned short	2	-32768..32767 0..65535
[int] или long или _int[32]	[signed] [int] unsigned [int] [signed] long unsigned long	4	$-2^{31}.. 2^{31}-1$ $0.. 2^{32}-1$
long long или _int[64]	[signed] long long Unsigned long long	8	$-2^{63}.. 2^{63}-1$ $0.. 2^{64}-1$
bool		1	false (0), true(1)

Примечание – Для совместимости считается: 0 – false; не 0 – true.

Вещественные типы

Тип	Размер, байт	Значащих цифр	Минимальное положительное число	Максимальное положительное число
float	4	6	1.175494351e-38	3.402823466e38
double (long double)	8	15	2.2250738585072014 e-308	1.797693134862318 e308

3. Неопределенный тип void

Нельзя объявлять значения типа void, он используется только при объявлении

- нетипизированных указателей;
- функций, не возвращающих значений (процедур).

Объявление переменных и поименованных констант

[<Изменчивость>] [<Тип>]<Список идентификаторов> [=<Значение>];

где <Изменчивость> – описатель возможности изменения значений:

- **const** – поименованная константа,
- **volatile** – переменная, меняющаяся в промежутках между явными обращениями к ней
- без указания изменчивости – обычная переменная

<Тип> – описатель типа: int, char, float, double и т.д.;

<Список идентификаторов> – список имен переменных или констант;

<Значение> – начальное значение переменной или значение

константы.

Примеры объявлений переменных и констант

Неинициализированные переменные:

```
int f,c,d; float r;  
I,j;unsigned int max,min;  
char c1,c2; unsigned char c5;
```

Инициализированные переменные

```
double k=89.34; char ch='G' ;
```

Поименованные константы

```
const long a=6; const float pp=6.6e-34;
```

На практике все объявления могут быть перемешаны в описаниях программы:

```
const char simt='T' ;float max=100,min=-100 ;  
double f,s,eps=0.001;
```

Переменные и поименованные константы могут быть объявлены в любом месте программы: вне всех функций, внутри функций, в любом месте функции.

Основное условие – объявление должно стоять до обращения к переменной или константе.

Перечисляемый тип

Используется для объявления набора поименованных целых констант.

Формат:

```
enum {<Ид>[=<Целое>] [,<Ид>[<>]...]}  
      <Список переменных>;
```

Пример:

```
enum {SUN, MON, TUES, FRI=5, SAT} day;
```

```
SUN = 0, MON = 1, TUES = 2, FRI=5,  
SAT=6
```

Имя
переменной

Константы присваиваются, начиная с нуля или с указанного значения.

Объявление типа пользователя

```
typedef <Описание типа> <Имя объявляемого типа>;
```

Примеры:

1) `typedef unsigned int word;`



Имя
НОВОГО ТИПА

2) `typedef enum {false, true} boolean;`



Имя
НОВОГО ТИПА