Типы данных

Типы данных определяют:

- область допустимых значений
- допустимые операции
- объём памяти
- формат хранения данных
- для предотвращения случайных ошибок

Базовые типы данных языка C

- **char** Используется как символьный тип и как однобайтовый целый.
- **int** Целый, знаковый тип.
- **float** Вещественный тип одинарной точности
- **double** Вещественный тип двойной точности
- **void** Пустой тип. Если функция имеет тип void, то она ничего не возвращает. Можно описать указатель на тип void, который может ссылаться на объект любого типа. Переменную типа void описать нельзя.

Модификаторы

Характеристики некоторых базовых типов могут быть изменены с помощью модификаторов:

- **Unsigned**
- **Signed**
- **Short**
- Long

Первые 3 модификатора применяются *только для целых типов*.

Long применяется для типа int (преобразует его к длинному целому 4 или 8 байт) и double (увеличивает его размер до 10 байт (в среде VS размер не изменяется))

Скалярные типы данных

Идентификатор типа	**Длина, байт**	**Диапазон значений**
Целые беззнаковые типы		
unsigned char	1	0..255
unsigned int	2 или 4	0..65535
unsigned short	2	0..65535
unsigned long	4	0..4294967295
Целые знаковые типы		
char	1	-128..127
int	2 или **4**	–32768…32767 или -2147483648…2147483647
short	2	–32768…32767
long	4	-2147483648…2147483647
Вещественные типы		
float	4	±(3.4E–38..3.4E+38)
double	8	±(1.7E–308..1.7E+308)
long double	10	±(3.4E–4932.. 3.4E+4932)

Типы данных языка Си++

- В языке Си++ поддерживаются все типы языка Си, а также добавлены новые типы.
- **Wchar_t**
- Расширенный символьный тип, предназначенный для работы с символами в кодировке Unicode. Занимает в памяти 2 байта. Для работы со строками типа wchar_t в библиотеке языка имеются отдельные версии функций обработки.
- **Bool**
- Логический тип с поддержкой констант false и true, занимающий в памяти 1 байт. При этом также поддерживаются все правила языка Си, касающиеся логических выражений

Константы

Это величины, не меняющие свои значения в программе

Целые константы

- Десятичные
- Восьмеричные (всегда начинается с нуля)
- Шестнадцатеричная (префикс 0x)

Константы имеют тип int или long. Если необходимо явно определить константу типа long, то добавляется суффикс L

123 = 0123 = 0x123 ?????? или 123 < 0123 < 0x123 ??????

Вещественные константы

- Все вещественные константы имеют тип double.

Пример 15.75; 1575e-2.

- Для описания именованных констант используется модификатор const.

Пример: const int z=100; const float pi=3.14;

- **Символьные константы** Это любой представимый или управляющий символ, заключенный в апостроф.

‘r’, ‘5’, ‘+’, ‘=‘

Значение символьной константы равно коду представляемого символа. Она может учувствовать в любых арифметических выражениях.

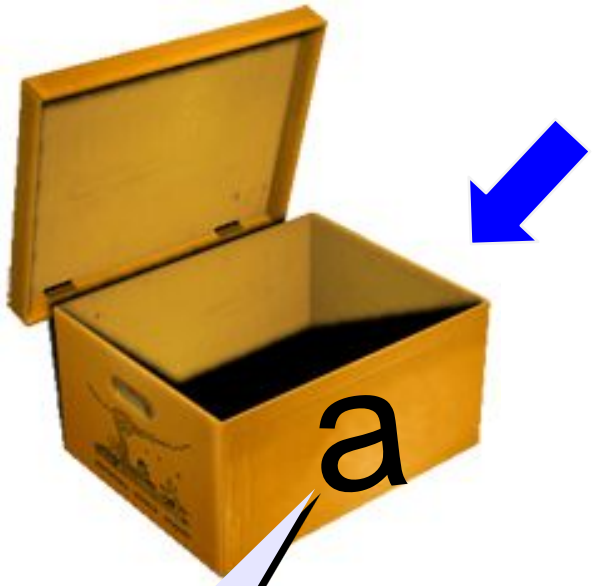
- **Строковые константы** Это последовательность символов (представимых или управляющих), заключенных в двойные кавычки.

“Скоро сессия“, “Группа 070017_“

Внутренне представление строковой константы – массив типа char. Количество выделяемых элементов равно количеству символов плюс один. Последним автоматически добавляется нуль-символ (‘\0’)

Переменные

- **Переменная** – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



Имя

Значение

Другой тип данных

! В переменной хранятся данные определенного типа!

Имена переменных

МОЖНО использовать

- латинские буквы (A-Z, a-z)

  заглавные и строчные буквы **различаются**

- цифры

  имя не может начинаться с цифры

- знак подчеркивания _

НЕЛЬЗЯ использовать

- ~~русские буквы~~
- ~~скобки~~
- ~~знаки +, =, !, ? и др.~~

Какие имена правильные?

AXby R&B 4Wheel Вася “PesBarbos”

TU154 [QuQu] _ABBA A+B

!! Переменная должна быть обязательно описана

Типы переменных:

- `int  // целая`
- `float  // вещественная`
- и другие…

Объявление переменных:

выделение места в памяти

тип – целые

список имен переменных

```
int a, b, c;
```

Описание может располагаться в любом месте программы. В описании указывается тип, за которым следуют переменные (пример: double a,b,c;).

Как записать значение в переменную?

оператор присваивания

```
a = 5;
```

! При записи нового значения старое стирается!

Оператор – это команда языка программирования (инструкция).

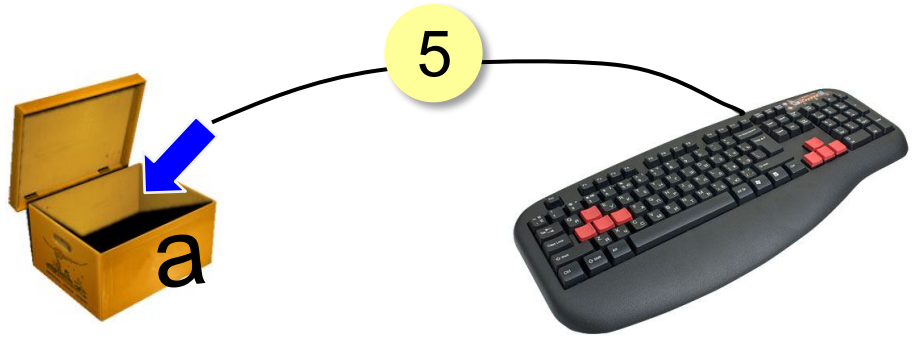
Оператор присваивания – это команда для записи нового значения в переменную.

При описании переменные могут быть инициализированы. Например: unsigned a=10, b=5;

Ввод значения с клавиатуры

ввести значение **a** из входного потока

```
cin >> a;
```



1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную **a**.

Ввод значений должен предваряться комментарием:

```
cout<<"Введи значение a";
cin >> a;
```

Несколько переменных вводятся через пробел или через *Enter*

```
cin >> a >> b;
```

Изменение значений переменной

```
int a, b;
a = 5;
b = a + 2;
a = (a + 2)*(b - 3);
b = b + 1;
```

Вывод данных

```
cout << a;
```

```
cout << a << endl;
```

```
cout << a << '\n';
```

```
cout << "Привет!";
```

```
cout << "Ответ: " << c;
```

```
//вывод текста и значения переменной c
```

```
cout << a << "+" << b << "=" << c;
```