**Переменные и типы данных**

При разработке и выполнении программы используются различные данные. Они могут быть входными, выходными или внутренними данными, которые используются временно.

В качестве данных могут выступать числа, символы, текст, массив чисел, массив строк, массив байтов, а также различные структуры данных, таких как очередь и пр.

Любые данные размещаются в памяти и хранятся в разном виде. Например, для числа может быть выделено четыре байта (32 разряда). Для одного символа – один байт. Структура хранения данных зависит от платформы вычислительного устройства и часто от системы программирования.

Для того чтобы определить структуру хранения данных в языке программирования вводится понятие **типа** данных. Все языки программирования различают основные типы данных:

1. Целые числа
2. Вещественные числа
3. Символы
4. Логический

В большинстве языков также имеется тип **строка**, хотя строку можно рассматривать как массив символов.

В большинстве языков имеется и более подробное деление типов, например, вещественное число с двойной точностью.

Стоит еще раз обратить внимание на то, что структура хранения данных зависит именно от типа. Т.е. если разные числа относятся к целым, то для них будет отводиться одинаковое количество памяти и структура их хранения тоже будет одинакова.

Программисты очень часто при работе с данными используют понятие **переменная.** Переменная – это именованный участок памяти, куда можно записывать или читать данные.

*Примечание. Существует также понятие* ***константа.*** *Ее можно один раз записать и затем только читать.*

**Объявление переменных**

Для того что бы в программе можно было использовать переменную, ее требуется **объявить**.

Именно для переменной объявляется тип данных. После такого объявления в переменную можно записывать или читать данные.

При объявлении, переменной присваивается имя и тип.

Имя обязательно должно начинается с буквы или цифры, но допускается и подчеркивание, а также некоторые символы, в зависимости от языка программирования.

Имя должно быть кратким и желательно смысловым. Не рекомендуется называть переменные А1, B3 и т.д. Все символы в имени должны быть английскими (кроме языка 1С Предприятие).

Во многих языках в имени имеет значение регистр, например, в С++ Т.е. имена MainName и mainame будут относится к разным переменным.

При объявлении каждой переменной обычно присваивается её тип. После объявления, операционная система выделяет место в памяти и определяет структуру хранения данных. Далее с переменной можно работать.

**Присваивание**

Объявление переменных – это подготовительный этап перед работой с переменными. Далее переменным, в том или ином виде, присваиваются конкретные значения: числа, символы и т.д. Часто переменой может быть присвоено содержание другой переменной или целого выражения.

В разных языках программирования операция присваивания может записываться по-разному. Но обычно **per=10**, где слева - название переменной, затем знак равенства и справа – значение, в соответствии с типом.

Часто совмещают объявление переменной и присваивание (инициализация):

**int per=10**

Далее все изложенное будем рассматривать на примерах в С++

**Основные типы данных в C++**

int — целочисленный тип данных.

float — тип данных с плавающей запятой.

char — символьный тип данных.

bool — логический тип данных.

**Пример объявления переменных в С++**

int num; // объявление переменной num целого типа.

float fnum; // объявление переменной fnum типа вещественного типа.

char sim; // объявление переменной sim символьного типа

bool log; // объявление логической переменной log.

**Пример присваивания значений переменным в С++**

num=10;

num2=1024;

fnum= 10.5;

sim=’b’;

log=true;

num=num2;

**Структура программы**

Как было указано ранее, программа — это описание алгоритма для решения определенной задачи. С другой стороны, программа – это язык общения программиста и компьютера. Следовательно, программа включает набор слов и некоторые правила употребления этих слов.

Количество слов в любом языке программирования не так много, и они различаются в написании: read, write, while, for, if и пр. А вот правил великое множество. Все языки программирования различаются именно правилами.

С++ не исключение. Поэтому со словами будем знакомится постепенно, а вот правила придётся разбирать более подробно.

Каждая программа на С++ имеет похожую структуру:

* **Правила работы данной программы**
* **Название и тип программы**
* **Начало программы**
* **Тело программы**
* **Конец программы**

**Правила работы данной программы** обычно включают заказ библиотек, которые используются при работе программы, используемые подпрограммы, описание классов и другие общие описания

**Название и тип программы** – в С++ и других с-подобных языках каждая программа кроме имени имеет и её тип. Упоминалось выше, что каждая переменная имеет тип, что определяет структуру хранения. Но в С++ и сама программа тоже может иметь тип: int, float и пр. Т.е. сама программа может выступать в роли переменной и использоваться в операции присваивания, например:

int num=raschet(); //num – переменная, raschet() – какая-либо

//внешняя программа

**Начало программы –** начало программы определяется символом {

**Тело программы** – состоит из команд С++ с помощью которых алгоритм и описывается. Часто команды называют операторами

**Конец программы –** определяется символом закрывающейся фигурной скобки }

**Пример программы на С++**

**Пример 1. Сложение чисел**

**#include <iostream> //библиотека для потокового ввода-вывода**

**using namespace std; //стандартное пространство имен**

**int main() //название и тип программы**

**{**

**float p; //объявление вещественной переменной p**

**int k; //объявление целой переменной k**

**float d; //объявление вещественной переменной d**

**p=1.2; //присвоили переменной p значение 1.2**

**k=5; //присвоили переменной k значение 5**

**d=p+k; //вычислили выражение p+k и результат**

**//записали в переменную d**

**cout << d; //выдали на экран значение переменной d**

**}**

p – переменная вещественного типа (дробное число)

к - переменная целого типа

d - переменная вещественного типа

p=1.2; переменной **p** присвоили значение 1.2

к=5; переменной **к** присвоили значение 5

По логике результат операции **p+k** должен быть равен **6.2**, но результат будет равен 6, т.к. они разного типа и дробная часть отбрасывается. Что бы результат был правильный, требуется изменить тип переменной к:

**#include <iostream> //библиотека для потокового ввода-вывода**

**using namespace std; //стандартное пространство имен**

**int main() //название и тип программы**

**{**

**float p; //объявление вещественной переменной p**

**float k; //объявление вещественной переменной k**

**float d; //объявление вещественной переменной d**

**p=1.2; //присвоили переменной p значение 1.2**

**k=5; //присвоили переменной k значение 5**

**d=p+k; //вычислили выражение p+k и результат**

**//записали в переменную d**

**cout << d; //выдали на экран значение переменной d**

**}**

В программу часто вставляют различные пояснения, которые называются комментариями. Компьютер «не видит» эти комментарии и тем более не пытается их выполнять как команды. Комментарии необязательны, но полезны для программиста. В С++ комментарии начинаются с символов //

**Пример 2. Сложение символьных значений**

**#include <iostream> //библиотека для потокового ввода-вывода**

**using namespace std; //стандартное пространство имен**

**int main() //название и тип программы**

**{**

**char p; //объявление символьной переменной p**

**char d; //объявление символьной переменной d**

**char s; //объявление символьной переменной s**

**p=’a’; //присвоили переменной p значение ‘a’**

**d=’d’; //присвоили переменной d значение ‘d’**

**s=’d’; //присвоили переменной s значение ‘d’**

**cout << p+d+s; //выдали на экран значение переменной d**

**}**

На экран должна быть выдана строка add, но этого не происходит. На экран будет выдано значение 297. Это произошло потому, что каждый символ представлен неким кодом. В данном случае в сумме участвовали эти коды.

Пример показывает, что в программировании существуют разные ограничения и программист должен их знать.

*Примечание. В некоторых языках можно складывать символы и результат будет адекватным. Например, в Java*

Программу можно немного сократить. Например,

**#include <iostream> //библиотека для потокового ввода-вывода**

**using namespace std; //стандартное пространство имен**

**int main() //название и тип программы**

**{**

**char p=’a’; //объявление символьной переменной p**

**char d=’d’; //объявление символьной переменной d**

**char s=’d’; //объявление символьной переменной s**

**cout << p+d+s; //выдали на экран значение переменной d**

**}**

Также будет выдано значение 297.

Можно добиться правильного результат следующим образом:

**#include <iostream> //библиотека для потокового ввода-вывода**

**using namespace std; //стандартное пространство имен**

**int main() //название и тип программы**

**{**

**char p=’a’; //объявление символьной переменной p**

**char d=’d’; //объявление символьной переменной d**

**char s=’d’; //объявление символьной переменной s**

**cout << p << d << s;**

**}**