

Лекция 2-3.

Язык программирования Си

1. Основные понятия языка Си.
2. Структура программы на Си.
3. Базовые типы данных в Си.

Учебное пособие стр. 23-37

Простейшая программа на Си

```
void main()  
{  
}
```

наша программа ничего не делает, она просто соответствует правилам языка Си, ее можно скомпилировать и получить `exe`-файл.

Что новенького?

```
#include <stdio.h>
void main()
{
printf ("Привет");
}
```

Эта программа выводит на экран слово
«Привет»

Лучше один раз увидеть!

```
1  #include <stdio.h>
2  #include <conio.h>
   using namespace std;
3  int main(void)
4  {
   int a,b,max;           //описание переменных
05  printf("a=");        //вывод приглашения ввести значение a
06  scanf("%d", &a);     //ввод значения переменной a
07  printf("b=");        //приглашение ввести значение b
08  scanf("%d", &b);     //ввод значения переменной b
09  if(a>b) max=a;       //если a>b то max=a
10  else max=b;         //иначе max=b
11  printf("max=%d\n", max); //вывод максимального значения
12  getch();           //останов до нажатия клавиши
13  return 0;          //выход из функции main
14  }
15
```

Алфавит языка

- Строчные и прописные буквы латинского алфавита
- |a|b|c| ... |x|y|z|A|B|C| ... |X|Y|Z|

- Арабские цифры

|0|1|2| ... |9|

- Специальные символы

|+|-|*|/|=|>|>=|<|<=|==|!|.|,|:|;|'|(|)|[|]|{| |}|&|@|\$|#|

- В алфавит языка Си также включается набор зарезервированных слов, имеющих строго определенное назначение. Например, слова **if**, **else**, **int**, **for** и др.

Основным назначением любой прикладной программы является преобразование исходных данных в соответствии с заданным алгоритмом.

Большая часть исходных данных обычно представлена числовой информацией.

Константы

- **Целые числа**

Например: 15 +1000 -47 02

- **Вещественные числа**

- Например: 2.5 -14.0 +0.33 0.0

- Числа с указанием порядка имеют вид: $a \cdot 10^p$, где a является мантиссой; p - десятичным

порядком

Математическая запись	Запись на языке Си
$3,14 \cdot 10^5$	3.14E5
$-17 \cdot 10^{-2}$	-17E-2
25,625	2.5625E1

Переменные

- Для обработки данных их необходимо хранить в памяти. При этом к этим данным надо как-то обращаться.
- в программировании каждой ячейке памяти (или группе ячеек) дается имя. Используя это имя можно прочитать информацию из ячейки и записать туда новую информацию.

Математическая запись	x	a_p	y^1	α	Σ	d_{-21}
Запись на языке Си	x	ap	y1	alpha	S	d_21

Переменные

- **Переменная** - это ячейка в памяти компьютера, которая имеет имя и хранит некоторое значение. Значение переменной может меняться во время выполнения программы. При записи в ячейку нового значения старое стирается.

Переменные, подобно константам, могут быть разных типов. Типы задаются стандартными зарезервированными словами:

integer - целый тип;

long – длинный целый тип;

short – целый тип с меньшим диапазоном;

float - вещественный тип;

double - вещественный тип с двойной точностью;

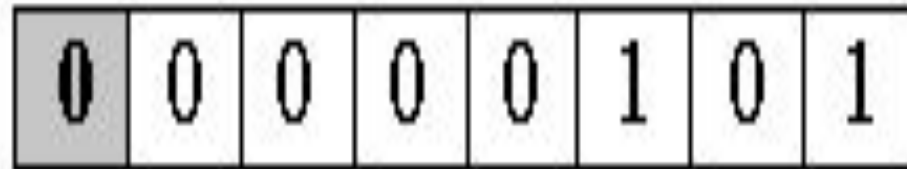
boolean - логический тип;

char - символьный тип;

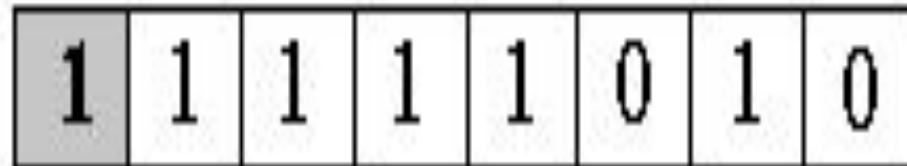
По описанию переменной в памяти компьютера резервируется ячейка для хранения значения переменной. В зависимости от объявленного типа переменной ячейка может иметь разную внутреннюю структуру, т.е. содержать различное число байт.

Внутреннее представление данных

+5



-5



Объявление и инициализация числовых переменных

```
const double eps=1E-6;  
void main(void)  
{  
    int i, j;  
    int x=18, y=-5;  
    float a=5.3;  
    ...  
}
```

Таблица базовых типов данных

Тип	Длина	Минимальное значение	Максимальное значение
char, signed char	1 байт	-128	127
unsigned char	1 байт	0	255
short int, short	2 байта	-32768	32767
unsigned short	2 байта	0	65535
int, signed	2 байта	-32768	32767
unsigned int, unsigned	2 байта	0	65535

Таблица базовых типов данных (продолжение)

long, long	4 байта	-2147483648	2147483647
int			
unsigned long	4 байта	0	4294967295
float	4 байта	3.4×10^{-38}	3.4×10^{38}
double	8 байт	1.7×10^{-308}	1.7×10^{308}
long double	10 байт	3.4×10^{-4932}	1.1×10^{4932}

Арифметические

выражения

- Арифметические выражения строятся из операндов, арифметических операций и круглых скобок.
- Операндами могут быть константы, переменные и функции.
- Арифметическое выражение является целым, если все входящие в него операнды целого типа. Результат вычисления целого арифметического выражения - целый.
- Если в арифметическом выражении содержится хотя бы один вещественный операнд, то результат - вещественный

Арифметические операции

* (умножение);

/ (деление);

% (выделение остатка от деления
целых чисел).

+ (сложение);

- (вычитание).

В бесскобочных арифметических
выражениях операции выполняются слева
направо в соответствии с их приоритетом.

Стандартные функции

хранятся в библиотеке (файл с именем math.h).

Запись на языке C	Математическая запись
Тригонометрические и обратные тригонометрические функции	
<code>sin(x)</code>	$\sin(x)$ - синус
<code>cos(x)</code>	$\cos(x)$ - косинус
<code>tan(x)</code>	$\operatorname{tg}(x)$ - тангенс
<code>asin(x)</code>	$\arcsin(x)$ - арксинус
<code>acos(x)</code>	$\arccos(x)$ - арккосинус
<code>atan(x)</code>	$\operatorname{arctg}(x)$ - арктангенс

Стандартные функции

(хранятся в библиотеке (файл с именем math.h).

Степени, экспоненты, логарифмы

<code>exp (x)</code>	e^x
<code>log (x)</code>	$\ln(x)$ – натуральный логарифм.
<code>log10 (x)</code>	– десятичный логарифм.
<code>sqrt (x)</code>	$\sqrt{\quad}$ - квадратный корень
<code>pow (x, y)</code>	x^y – возведение x в степень y

Стандартные функции

(хранятся в библиотеке (файл с именем `math.h`).

Модуль, знак, минимальное, максимальное число	
<code>abs (m)</code>	Абсолютное значение числа. Аргумент типа <code>int</code> , <code>long</code> , <code>float</code> или <code>double</code> . Результат того же типа, что аргумент.
<code>min (x, y)</code>	Минимальное из двух чисел. Аргументы одного типа. Возможны типы: <code>int</code> , <code>long</code> , <code>float</code> , <code>double</code> . Результат того же типа, что аргумент.
<code>max (x, y)</code>	Максимальное из двух чисел. Аргументы одного типа. Возможны типы: <code>int</code> , <code>long</code> , <code>float</code> , <code>double</code> . Результат того же типа, что аргумент.

Стандартные функции

(хранятся в библиотеке (файл с именем

Округления

`ceil(x)`

Ближайшее к **x** целое,
больше или равно **x**

`floor(x)`

Ближайшее к **x** целое,
меньше или равно **x**

`round(x)`

Ближайшее к **x** целое.
Аргумент типа `float` или
`double`. Результат типа `long`,
если аргумент `double`, и типа
`int` – если `float`. То же, что
`(int) floor(x + 0.5)`.

Стандартные функции

(хранится в библиотеке (файл с именем stdlib.h)).

Случайное число

`rand(n)`

Псевдослучайное
число в диапазоне от
0 до n.

Примеры программирования арифметических выражений

Математическая запись	Запись на языке Си
$\frac{a + 12b}{c_1 - 1,8 \cdot 10^3}$	<code>(a+12*b)/(c1 - 1.8E3)</code>
$e^{\frac{2 \sin 4x + \cos^2 x^2}{3x}}$	<code>exp((2*sin(4*x)+pow((cos(x*x),2)))/(3*x))</code>
$\ln \operatorname{tg} \alpha - \sin \alpha^3 $	<code>log(abs(tan(a1)-sin(pow(a1,3))))</code>

Оператор присваивания

- Оператор присваивания служит для вычисления выражения и записи результата в память компьютера.
- Общий вид записи оператора

имя переменной = выражение ;

- Знак = читается как «присвоить». Конец любого оператора на языке Си фиксируется точкой с запятой.

Примеры оператора присваивания

```
Y=exp ( (2*sin (4*x) +pow ( (cos (x*  
x) , 2) ) ) / (3*x) ) ;
```

```
S= (a+12*b) / (c1 - 1.8E3) ;
```


Пример программы

```
#include <stdio.h>
#include <math.h>
void _tmain()
{ float Y,S,x,a,b,c1; //создание переменных
printf("vvedite ishodnie dannie x,a,b,c1 ");
scanf("%f%f%f%f",&x,&a,&b,&c1); /*ввод
исходный данных */
Y=exp( (2*sin(4*x)+pow( (cos(x*x),2)) ) / (3*x) );
S=(a+12*b) / (c1 - 1.8E3);
printf("Rezultat:Y= %f\n", Y); //вывод значения Y
printf("Rezultat: S=%f\n",S); //вывод значения S
} /*Эта программа вычисляет выражения для Y и S и
выводит на экран их значения */
```