

Лекция 2. Переменные и операции.

Евгений Линский

Вывести на экран "Hello, world!"

```
#include <stdio.h>

int main() { // идентификатор main
    printf("Hello, world!"); //вывод на экран
    return 0;
}
```

- ▶ В любой программе на C должен быть такой "main."
- ▶ Процессор начинает исполнение программы с него и последовательно исполняет все строки.
- ▶ Операторы программы должны находиться внутри скобок { }.
 - По одному на строке
 - Завершаются с помощью ;

```
int a;  
int a, b, c; // Error: имя a уже использовано  
int d = 1, k = 0; // Задано начальное значение  
                (инициализация)
```

- ▶ Переменная — именованная область памяти
 - В памяти зарезервировали свободную ячейку
 - Дали ячейки имя (чтобы не думать про адреса)
- ▶ Формат: тип имя; (int a;). Тип задает:
 - размер ячейки (сколько байт зарезервировано)
 - правила работы с ячейкой (целые числа/вещественные числа):

Примитивные типы данных

В памяти машины переменная хранится в виде соответствующей последовательности единиц и нулей (бит).

Название типа	Кол-во байт для хранения	Диапазон
char	1	$-2^7 \dots 2^7 - 1$
short	2	$-2^{15} \dots 2^{15} - 1$
int	4	$-2^{31} \dots 2^{31} - 1$
long	4	$-2^{31} \dots 2^{31} - 1$
long long	8	$-2^{63} \dots 2^{63} - 1$
unsigned char	1	$0 \dots 2^8 - 1$
unsigned short	2	$0 \dots 2^{16} - 1$
unsigned int	4	$0 \dots 2^{32} - 1$
unsigned long	4	$0 \dots 2^{32} - 1$
unsigned long long	8	$0 \dots 2^{64} - 1$
float	4	$1,4 \cdot 10^{-45} \dots 3,4 \cdot 10^{38}$
double	8	$4,94 \cdot 10^{-324} \dots 1,79 \cdot 10^{308}$

- ▶ float, double — вещественные числа
- ▶ остальные — целые числа

Программу чаще читают чем пишут!

```
int a = 0;
int b = 2;
int c = 10;
// При каждом запуске программы в d новое произвольное значение.
int d;
// Память не очищается!
```

- ▶ Код удобнее редактировать и читать, когда каждая переменная объявляется на отдельной строке.
- ▶ Желательно при этом сразу задавать начальное значение.

```
int a = 10; //десятеричная система
//шестнадцатеричная (запись числа начинается с 0x)
int b = 0xA;
//восьмеричная (запись числа начинается с 0)
int c = 012;
```

Операции с целочисленными переменными

```
int a = 3;  
int b = 5;  
int c = 0;  
// = это оператор "присвоить"  
c = a + b; // сложить a и b и записать результат в c  
c = a + 10;
```

Можно написать и так:

```
int a = 3;  
a = a + 5; // теперь a записано число 8
```

Смысл: взять значение из *a*, прибавить к нему 5 и опять записать в *a*.

Какие еще бывают операции?

Операции: +, -, *, /, % (взятие остатки)

```
int a = 3;
int b = 5;
int c = 0;
int d = 0;
c = b / a; // деление нацело: в c запишется 1
d = b % a; // остаток от деления: в c запишется 2
```

Можно записать арифметическое выражение:

```
int a = 3;
int b = 5;
int c = 8;
int d = 0;
d = (b * b - 4 * a * c) / (2 * a);
```


- ▶ Программисты любят все сокращать!
- ▶ Так быстрее читать и писать.

```
int a = 5;  
a += 3;    // a = a + 3;  
a++;      // a = a + 1;
```

Еще бывает:

```
int x = 4;  
a -= x;  
a *= x;  
a /= x;  
a %= 3;  
a --;
```

Обмен значений двух переменных

С использованием третьей переменной.

```
int a = 5;  
int b = 3;  
int tmp = 0;  
tmp = a;  
a = b;  
b = tmp;
```

Q: как обменять значения двух переменных, не используя третью?

```
#include <stdio.h> // необходимо для работы ввода/вывода

int main() {
    int a = 3;
    int b = 0;

    printf("%d \n", a); // вывести экран целое число из a
                        // и перевести строку

    scanf("%d", &b); // считать с клавиатуры число
                    // и записать его в b

    return 0;
}
```

- ▶ Не забудьте & при вводе!
- ▶ *printf*, *scanf* — f означает “форматный”
 - задаем формат, того что хотим вывести или считать
 - %d – хотим целое число в десятичной системе счисления

```
int a = 3;
int b = 5;
int c = 8;
int d = 0;
scanf("%d", &a);
d = (b * b - 4 * a * c) / (2 * a);
```

Что может пойти не так?

```
int a = 3;
int b = 5;
int c = 8;
int d = 0;
scanf("%d", &a);
d = (b * b - 4 * a * c) / (2 * a);
```

Что может пойти не так?

- ❶ На ноль делить нельзя!

```
int a = 3;
int b = 5;
int c = 8;
int d = 0;
scanf("%d", &a);
d = (b * b - 4 * a * c) / (2 * a);
```

Что может пойти не так?

- 1 На ноль делить нельзя!
- 2 Если пользователь введет 0, то программа аварийно завершится.