

Основы программирования. Занятие №1



Как все работает?

- Принцип действия большинства современных устройств состоит во взаимодействии двух сред – аппаратной и цифровой:
- Первая – это набор микросхем, датчиков и других физических элементов, которые необходимы для корректного функционирования устройства.
- Вторая - набор инструкций(программного кода) для управления процессом работы.



Программирование

- Процесс написания упорядоченного набора инструкций для аппаратуры устройства. Именно они указывают процессору и всем связанным с ним деталям, какие вычисления им необходимо производить и в какой последовательности.

```
252     ...updatePhotoDescription( cell ) {
253     }
254
255     function updatePhotoDescription() {
256     if (descriptions.length > (page * 9) + (currentImage.substring(0, 1))) {
257     document.getElementById( 'bigimageDesc' ).innerHTML = descriptions[page * 9 + currentImage.substring(0, 1)];
258     }
259     }
260
261     function updateAllImages() {
262     var i = 1;
263     while (i < 10) {
264     var elementId = 'foto' + i;
265     var elementIdBig = 'bigimage' + i;
266     if (page * 9 + i - 1 < photos.length) {
267     document.getElementById( elementId ).src = 'images/' + photos[page * 9 + i - 1];
268     document.getElementById( elementIdBig ).src = 'images/' + photos[page * 9 + i - 1];
269     } else {
270     document.getElementById( elementId ).src = '';
271     }
```



Работа компьютера



Виды программирования

- Низкоуровневое – близкое к программированию непосредственно в машинных кодах используемого реального или виртуального процессора.
- Высокоуровневое – основная черта - это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде очень длинны и сложны для понимания.



Программа “Hello world”

Низкоуровневое

```
.MODEL SMALL
.STACK 100h
.DATA
    HelloMessage DB 'Hello World',13,10,'$'
.CODE
START:
    mov ax,@data
    mov ds,ax
    mov ah,9
    mov dx,OFFSET HelloMessage
    int 21h
    mov ah,4ch
    int 21h
END START
```

Высокоуровневое

```
#include <stdio.h>

void main() {
    printf("Hello, World!\n");
}
```



Как работают программы

- 1. Помещение приложения в оперативную память.
- 2. Передача управления приложению(точка входа).
- 3. Выполнение инструкций приложения.

```
#include <stdio.h>

void main() {
    printf("Hello, World!\n");
}
```



Вывод данных - Функция printf

- Функция для вывода информации на стандартный вывод(напр. консоль).
- Для работы необходимо подключить библиотеку стандартного ввода/вывода(«stdio.h»).

```
#include <stdio.h>

void main() {
    printf("Hello, World!\n");
}
```



Как хранятся данные?

- Двоичная система счисления - В двоичной системе счисления числа записываются с помощью двух символов (0 и 1).
- В n битах может содержаться 2^n значений.
- Сколько значений в 8 битах?



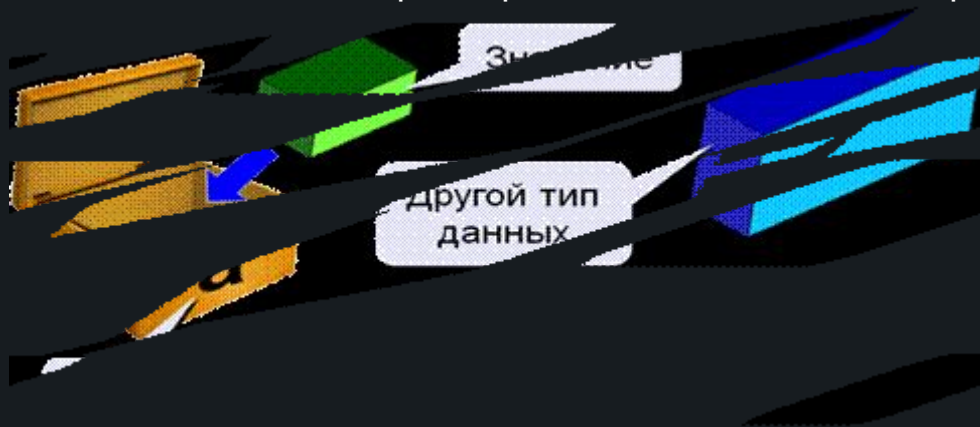
Типы данных

Тип	Размер в байтах	Диапазон значений
char	1	от -128 до 126
unsigned char	1	от 0 до 255
short	2	от -32 768 до 32 767
unsigned short	2	от 0 до 65536
enum	2	от -2 147 483 648 до 2 147 483 647
long	4	от -2 147 483 648 до 2 147 483 647
unsigned long	4	от 0 до 4 294 967 295
int	4	от -2 147 483 648 до 2 147 483 647
unsigned int	4	от 0 до 4 294 967 295
float	4	от $3,4 * 10^{-38}$ до $3,4 * 10^{38}$
double	8	от $1,7 * 10^{-308}$ до $1,7 * 10^{308}$
long double	10	от $3,4 * 10^{-4932}$ до $1,1 * 10^{4932}$
bool	1	true или false



Переменная

- Именованная область памяти с данными.
- В высокоуровневых языках имеет размер в зависимости от типа хранимых данных.



Переменная в Си

- Объявление: **Тип Имя = Значение;**
- Пример: `int a = 2;`
- **Над числовыми типами данных доступны основные арифметические операции (+, -, *, /)**

```
int a = 1;  
int b = 2;  
int c = 0;  
c = a + b;  
c = a - b;  
c = a * b;  
c = a / b;
```



Вывод данных - Функция printf

- Для вывода чисел используется форматированный вывод. В строку вставляется идентификатор параметра, а последующими параметрами идут сами значения.
- %d – целое число.

```
int x = 2;  
int y = x * x + x + 5;  
printf("Result = %d", y);
```

```
int x = 2;  
int y = x * x + x + 5;  
printf("x = %d, y = %d", x, y);
```



Практика

- Посчитать выражение. $x * x + x + 5$



Домашнее задание

- Посчитать выражение. $5 * x - (x - 5)$



Что установить?

- <http://www.jetbrains.com/clion/> - CLion от JetBrains
- <https://www.visualstudio.com/ru/vs/community/> - Visual Studio от Microsoft

