

## 1.1. Первая программа.

### Вывод текста и комментирование кода.

Wednesday, March 11, 2020

20:28



main

## Hello world!

Как и в случае с микроконтроллерами, начнем с написания простейшей программы. Здесь наипростейшей программой считается вывод в консоль какой-нибудь фразы, часто "Hello world!". Запишем код и разберем его.

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     printf("Hello world!\n");
6
7     return 0;
8 }
```

Итак, по порядку. Во-первых, в С важен регистр букв, поэтому надо за ним следить.

Например, `Int` и `int` - совершенно разные вещи с точки зрения языка. Основной файл может называться как угодно, но общепринятое название – `main`. Расширение обязательно должно быть `.c`

Теперь перейдем к разбору кода.

Первая строка подключает библиотеку `stdio.h`:

```
#include <stdio.h>
```

Она обычно будет нужна во всех программах на С, потому что в самом языке С нет простых способов осуществить ввод-вывод, а эта библиотека содержит функции для ввода-вывода данных, на что и намекает ее название. Функция `printf` также является ее частью и не будет работать, если мы не подключим эту библиотеку. Также обратите внимание на отсутствие точки с запятой после директивы `#include`.

Со строки

```
int main(void)
```

начинается выполнение программы. Это стандарт в С, отклоняться от которого **недопустимо** [2].

Основная функция должна называться `main` и возвращать значение типа `int`. Слово `void` означает, что функция не принимает на вход никакие значения. В некоторых случаях функция `main` может выглядеть следующим образом:

```
int main(int argc, char *argv[])
```

В нашем контексте эти функции идентичны и разницу между ними мы рассмотрим позже.

Строка

```
printf("Hello world!\n");
```

выводит текст в скобках в консоль. Символ `\n` дает команду на перенос курсора на новую строку. Как `println` в паскале. Без него тоже будет работать, но так красивее выглядит, наверное.

И, наконец, последняя строка

```
return 0;
```

завершает функцию `main()` и возвращает операционной системе значение типа `int`. Вместо 0 может быть любое значение типа `int`. Программисты договорились, что возвращение 0 означает, что программа завершилась без ошибок. Разными числами можно возвращать разные типы ошибок (вспомните, как разные программы часто выдают ошибки с номером).

Выведем несколько строк по порядку:

```
printf ("Testing...\n..1\n...2\n....3\n");
```

В результате на выходе получим:

```
Testing...
...1
...2
....3
```

Теперь попробуем вывести значения каких-нибудь переменных. Объявим три целочисленных (integer) переменных – `value1`, `value2` и `sum`, подставим их в простую формулу и выведем результаты.

```
1 #include <stdio.h>
2
3 int main (void)
4 {
5     int value1, value2, sum;
6
7     value1 = 50;
8     value2 = 25;
9     sum = value1 + value2;
10    printf ("The sum of %i and %i is %i.\n", value1, value2, sum);
11
12    return 0;
13 }
```

На выходе получим:

```
The sum of 50 and 25 is 75.
```

Как видим, параметр `%i` подставляет значение переменной той же позиции в списке, что и параметр в строке.

## Комментирование кода

Есть два способа комментировать код:

```
1 /* комментарий1
2    комментарий 2
3 */
```

и

```
// комментарий
```

Первый вариант обычно используется когда нужно вставлять большие блоки комментариев. Второй вариант работает в пределах одной строки. Следует обратить внимание, что если открыть комментарий `/*` и не закрыть его `*/`, то будет закомментирован весь текст, который находится после открывающего оператора.

Давайте закомментируем наш код:

```
1 /* This program adds two integer values
2    and displays the results */
3
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Declare variables
9     int value1, value2, sum;
10
11    // Assign values and calculate their sum
12    value1 = 20;
13    value2 = 50;
14    sum = value1 + value2;
15
16    // Display the result
17    printf ("The sum of %i and %i is %i.\n", value1, value2, sum);
18 }
```

```
19     return 0;  
20 }
```

Теперь код гораздо удобнее читать. Главное – не увлекаться и не комментировать очевидное, как мы только что сделали.

---

#### Источники:

1. Stephen Kochan – Programming in C (4th Edition); chapter 2
2. <https://stackoverflow.com/questions/449851/why-do-we-need-to-use-int-main-and-not-void-main-in-c>
3. K.N. King – C Programming: A Modern Approach, 2nd Edition; chapter 2