

# Информатика

Язык C/C++  
осень 2017

Информатика

Кочанов Марк



Пример #1

Пример #2

Арифметика

Операции  
сравнения

Оператор if...else

Циклы

Задачи

Кочанов Марк  
НИЯУ МИФИ

# Пример #1

## Листинг

```
1  /* the first program
2     the second line */
3  #include <stdio.h>
4
5  int main(void) {
6      printf("Hello, world\n"); // write in console
7
8      return 0;
9  }
```

```
Hello, world
```



# Пример #1

## Комментарии

```
1  /* the first program
2     the second line */
3  #include <stdio.h>
4
5  int main(void) {
6      printf("Hello, world\n"); // write in console
7
8      return 0;
9  }
```

В языках C/C++ доступны два варианта комментариев:

- ❶ многострочные комментарии `/* */` (C style)
- ❷ однострочные комментарии `//` (C++ style), доступны в языке C, начиная со стандарта C99<sup>1</sup>

---

<sup>1</sup>Стандарт ISO/IEC 9899:1999 был выпущен в 1999 году. Первый стандарт языка C известен как ANSI C, также известный как C89 и C99. Последний на данный момент стандарт C выпущенный в 2011 году.



# Пример #1

## Заголовочные файлы

```
1  /* the first program
2     the second line */
3  #include <stdio.h>
4
5  int main(void) {
6     printf("Hello, world\n"); // write in console
7
8     return 0;
9 }
```

Директива **#include** необходима для подключения соответствующего заголовочного файла. Заголовочный файл `stdio.h` содержит определения функций ввода/вывода, таких как `printf()` и `scanf()`.



# Пример #1

## Функция main()

```
1  /* the first program
2     the second line */
3  #include <stdio.h>
4
5  int main(void) {
6     printf("Hello, world\n"); // write in console
7
8     return 0;
9 }
```

Функция main() служит точкой входа в программу (с неё начинается исполнение программы).

Каждая программа должна содержать только одну точку входа.

Функция main() должна вернуть целое число<sup>2</sup>. В случае успешного завершения ноль.

---

<sup>2</sup>начиная с C99 (<https://stackoverflow.com/questions/204476/what-should-main-return-in-c-and-c>)



# Пример #1

## Вывод в консоль

```
1  /* the first program
2     the second line */
3  #include <stdio.h>
4
5  int main(void) {
6      printf("Hello, world\n"); // write in console
7
8      return 0;
9  }
```

Функция `printf()` используется для форматированного вывода информации в консоль.

Комбинация `\n` необходима для перехода на новую строку и перевода курсора на начало строки.

Каждая команда должна заканчиваться точкой с запятой.



# Пример #2

## Листинг

```
1  #include <stdio.h>
2
3  int main(void) {
4      int int1, int2;
5      int sum = 0;
6
7      printf("Enter first integer\n");
8      scanf("%d", &int1);
9
10     printf("Enter second integer\n");
11     scanf("%d", &int2);
12
13     sum = int1 + int2;
14     printf("Sum is %d\n", sum);
15
16     return 0;
17 }
```

```
Enter first integer
23
Enter second integer
10
Sum is 33
```



# Пример #2

## Листинг

```
4      int  int1, int2;  
5      int  sum = 0;
```

В языках C/C++ любая переменная должна быть объявлена (declaration) перед первым использованием.

Переменная `sum`, кроме того, определена (definition).

Для любой переменной возможно одно объявление и любое количество определений. Переменные `int1` и `int2` также называются неинициализированными. Обращение не к инициализированным переменным приводит к неопределенному поведению (undefined behaviour).

Имя (идентификатор) состоит из последовательности букв и цифр. Знак `_` подчеркивания считается буквой. Объявление должно начинаться с буквы, однако использовать для этих целей знак подчеркивания не рекомендуется.

Символы верхнего и нижнего регистра различаются. Языки C/C++ регистрозависимы!





# Пример #2

## Ввод/вывод с консоли

```
8 scanf("%d", &int1);
```

Оператор `scanf()` используется для считывания данных со стандартного устройства ввода. `%d` (format specifier) является escape-последовательностью и задает формат считывания (целое число).

Амперсанд `&` служит для взятия адреса переменной.

```
14 printf("Sum is %d\n", sum);
```

Escape-последовательность `%d` (format specifier) необходима для указания типа выводимой переменной.

Количество «спецификаторов формата» не должно превышать количества передаваемых переменных.





Действие	Операция	Выражение
умножение	*	$a * 10$
деление	/	$1.0 / a$
взятие по модулю	%	$a \% 2$
сложение	+	$a + 5$
вычитание	-	$a - b$

Операции в таблице расположены в порядке понижения приоритета. Приоритет может быть изменен при помощи скобок  $()$ .

Выражения вычисляются слева направо.



Пример #1

Пример #2

Арифметика

Операции  
сравнения

Оператор if...else

Циклы

Задачи

	Операция	Выражение
равно	<code>==</code>	<code>x == 5</code>
не равно	<code>!=</code>	<code>x != y</code>
больше	<code>&gt;</code>	<code>x &gt; 10</code>
меньше	<code>&lt;</code>	<code>x &lt; y</code>
больше или равно	<code>&gt;=</code>	<code>x &gt;= 5</code>
меньше или равно	<code>&lt;=</code>	<code>x &lt;= y</code>

Тип данных `bool` доступен начиная с версии C99 и объявляется в заголовочном файле `stdbool.h`. Переменная типа `bool` может принимать два значения: `true` и `false`.

По определению, при преобразовании к типу `int` значению `true` сопоставляется 1, а значение `false` — 0. И наоборот: ненулевые значения трактуются как `true`, в 0 как `false`.

# Условный оператор if...else



Пример #1

Пример #2

Арифметика

Операции  
сравнения

Оператор if...else

Циклы

Задачи

```
1  if ( /* expression */ )
2      /* statement_true */
3
4  if ( /* expression */ )
5      /* statement_true */
6  else
7      /* statement_false */
8
9  int i = 2;
10 if (i > 2) {
11     printf("first is true\n");
12 } else {
13     printf("first is false\n");
14 }
15
16 i = 3;
17 if (i == 3)
18     printf("i == 3\n");
19
20 if (i != 3)
21     printf("i != 3 is true\n");
22 else
23     printf("i != 3 is false\n");
```

# Вложенный if и else if



```
1 if (a > 0)
2     printf("A is greater than zero\n");
3 else if (a < 0)
4     printf("A is lesser than zero\n");
5 else
6     printf("A is zero\n");
```

```
1 if ( a >= 0 ) {
2     if ( a > 0 )
3         printf("A is greater than zero\n");
4     else
5         printf("A is zero\n");
6 } else
7     printf("A is lesser than zero\n");
```

# Циклы

## Цикл for

```
1  for ( /* init_clause */; /* cond_exp */; /* iteration_exp */ )
2      /* loop_statement */
3
4  for(int n = 0; n < 10; ++n, )
5      printf("%d\n", n);
6
7  for(;;) {
8      printf("endless loop!");
9  }
```

- `init_clause` вычисляется один раз, в самом начале
- `cond_expression` вычисляется перед каждой итерацией цикла. Если результат выражения 0 или `false`, то цикл обрывается



# Циклы

## break и continue

Для прерывания цикла может быть использован оператор break

```
1  n = 0;
2  for(;;) {
3      printf("endless loop with break!");
4      ++n;
5
6      if (n >= 5)
7          break;
8  }
```

Для прерывания текущей итерации цикла применяется оператор continue

```
1  for (int i = 0; i <= 5; i++) {
2      if ( i % 2 == 1 )
3          continue;
4
5      printf("%d is even\n", i);
6  }
```



# Циклы

## Цикл while (while loop)

```
1  while ( /* expression */ )
2      /* statement */
3
4  n_max = 5;
5  n = 0;
6  while(n < n_max) {
7      printf("%d ", n);
8      ++n;
9  }
```

Условие expression проверяется **перед** выполнением выражения (итерацией цикла).

Для цикла while могут быть использованы операторы break и continue.





# Циклы

## Цикл do while

```
1  do /* statement */
2  while ( /* expression */ );
3
4  n_max = 5;
5  n = 0;
6  do {
7      printf("%d ", n);
8      ++n;
9  } while (n < n_max);
```

Условие expression проверяется **после** выполнения итерации. Для цикла while могут быть использованы операторы break и continue.



# Задача 1



Пример #1

Пример #2

Арифметика

Операции  
сравнения

Оператор if...else

Циклы

Задачи

Пользователь с клавиатуры вводит неотрицательное целое число  $N$ . Затем вводит последовательно  $N$  целых чисел.

Вывести на экран среднее арифметическое.

```
4
-10
3
5
10
Arithmetic mean is 4
```

Для хранения среднего арифметического использовать тип данных float (дробное число).

## Задача 2



Пример #1

Пример #2

Арифметика

Операции  
сравнения

Оператор if...else

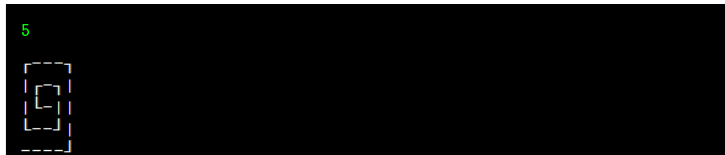
Циклы

Задачи

Пользователь с клавиатуры вводит целое положительное число  $N$  больше или равное 3. Нарисовать квадратную спираль ( $N$  строк и столбцов), закрученную внутрь по часовой или против часовой стрелки и начинающуюся в одном из 4 углов. Для четного номера варианта по часовой стрелке, для нечетного — против часовой стрелки. Если остаток от деления номера варианта на 4 равен

- 0, то спираль должна начинаться в левом верхнем углу
- 1, то спираль должна начинаться в правом верхнем углу
- 2, то спираль должна начинаться в правом нижнем углу
- 3, то спираль должна начинаться в левом нижнем углу

7 вариант:



Для хранения среднего арифметического использовать тип данных float (дробное число).

## Задача 2 (подсказки)



Для вывода «уголков» необходимо использовать спецификатор формата %c, а после запятой указать ASCII-код нужного символа.

```
1 printf("%c%c\n", 218, 191);  
2 printf("%c%c\n", 192, 217);
```