

🏠 (<http://cppstudio.com>)

/ Стандартные заголовочные файлы из Си в C++ (<http://cppstudio.com/cat/309/>)

/ Заголовочный файл cstdio (stdio.h) (<http://cppstudio.com/cat/309/323/>) / Функция fprintf

Функция fprintf



Оценка: **3,50** (голосов: 2)

Чтобы проголосовать, вы должны зарегистрироваться.

Прототип функции fprintf:

```
1 int fprintf( FILE * stream, const char * format, ... );
```

Заголовочный файл

Название	Язык
stdio.h	C
cstdio	C++

Описание

Функция `fprintf` выполняет форматированный вывод в поток. Записывает в указанный поток последовательность символов в формате, указанном аргументом `format`. После параметра `format`, функция ожидает, по крайней мере, многие дополнительные аргументы, как указано в прототипе.

Параметры:

- **stream**

Указатель на объект типа `FILE`, который связан с потоком.

- **format**

Си-строка, содержащая текст, который будет выведен на поток. Опционально, строка может содержать встроенные метки форматирования, которые заменяются значениями, указанными в последующих дополнительных аргументах и отформатированы требуемым образом.

Форматируемые теги должны следовать следующему формату записи:

```
%[флаги][ширина поля][.точность][длина]спецификатор
```

Ниже приведены спецификаторы форматирования потоков ввода/вывода:

Спецификатор	Описание	Пример
c	Символ	d
d или i	Знаковое десятичное число.	111
e	Экспоненциальная форма записи чисел (мантисса/экспонента) с использованием символа e.	3.9265e+2
E	Экспоненциальная форма записи чисел (мантисса/экспонента) с использованием символа E.	3.9265E+2
f	Десятичное значение с плавающей точкой.	392.65
g	Используется для обозначения короткого %e или %f	392.65
G	Используется для обозначения короткого %E или %f	392.65
o	Восьмеричная система счисления	610
s	Строка символов.	sample
u	Использовать беззнаковое целое десятичное значение.	7235
x	Использовать беззнаковые шестнадцатеричные значения	7fa
X	Шестнадцатеричные целые без знака (прописные буквы)	7FA
p	Указатель на адрес	B800:0000
n	Ничего не печатается. Аргумент должен быть указателем на знаковое целое, где хранится количество символов, написанных до сих пор.	
%	Следующий символ, который должен быть записан в поток.	

Форматирующие теги могут содержать флаги, ширину, точность и модификаторы, которые являются необязательными:

Флаг	Описание
–	Выравнивание по левому краю выделенного поля. По умолчанию стоит правостороннее выравнивание в выделенном поле.
+	Использовать знаки плюс или минус для вывода значений. По умолчанию выводится только знак минус, если значение отрицательное.
(space)	Если в строке есть символ пробела, он не игнорируется, а попадает в поток вывода.

#	Используется вместе со спецификаторами <code>o</code> , <code>x</code> или <code>X</code> специфицирующих значение отличное от нуля, <code>0x</code> или <code>0X</code> . Используется вместе со спецификаторами <code>e</code> , <code>E</code> и <code>f</code> , принудительно выводит десятичную точку, даже, если у значения нет цифр в дробной части. По умолчанию, если нет цифр в дробной части, десятичная точка не отображается. Используется с <code>a</code> , <code>A</code> , <code>e</code> , <code>E</code> , <code>f</code> , <code>F</code> , <code>g</code> или <code>G</code> результат такой же, как и со спецификаторами <code>e</code> или <code>E</code> , но нули не удаляются.
0	Левая часть после числа заполняется нулями (0) вместо пробелов, если ширина поля больше числа.

ширина	Описание
<i>number</i>	Минимальное количество символов для печати. Если значение для печати короче, чем это число, пустые места заполняются пробелами. Значение не обрезается, даже если оно больше <i>number</i> .
*	Ширина не указана в строке формата, но в качестве дополнительного аргумента целое значение предшествующего аргумента, который должен быть отформатирован.

.точность	Описание
<i>.number</i>	Для спецификаторов целых чисел <code>d</code> , <code>i</code> , <code>o</code> , <code>u</code> , <code>x</code> , <code>X</code> : <i>указывает минимальное количество цифр</i> . Если значение, короче, чем это число, пустое место заполняется нулями. Значение не обрезается, даже если <i>number</i> короче. Точность 0 означает, что нет символа записанного для значения 0. Для спецификаторов <code>e</code> , <code>E</code> и <code>F</code> : это количество цифр, которое будет напечатано после десятичной точки. Для спецификаторов <code>g</code> и <code>G</code> : это максимальное количество значащих цифр для печати. Для <code>s</code> – это максимальное количество символов для печати. По умолчанию все символы выводятся пока не встретится нулевой символ. Для <code>c</code> : этот параметр не имеет значения. При отсутствии этого параметра, точность по умолчанию равна 1. Если срок указан без явного значения для точности, значение 0.
.*	Точность не указана в строке формата, но в качестве дополнительного аргумента целое значение предшествующего аргумента, который должен быть отформатирован.

длина	Описание
h	Короткий <code>int</code> или беззнаковый короткий <code>int</code> (применяется только к целым спецификаторам: <code>i</code> , <code>d</code> , <code>o</code> , <code>u</code> , <code>x</code> и <code>X</code>).
l	Длинный <code>int</code> или длинный беззнаковый <code>int</code> , применяется к целым спецификаторам (<code>i</code> , <code>d</code> , <code>o</code> , <code>u</code> , <code>x</code> и <code>X</code>), а также к широким символам или строкам с широкими символами, для спецификаторов <code>c</code> и <code>s</code> .

Дополнительные аргументы

В зависимости от формата строки, функция может принимать дополнительные аргументы, каждый из которых содержит одно значение. Вместо каждого %-тега, указанного в параметре `format`, в поток вывода будет вставлено значение соответствующего аргумента. Количество дополнительных аргументов должно соответствовать количеству %-тегов.

Возвращаемое значение

В случае успеха, возвращается общее число записанных символов.

В случае неудачи, возвращается отрицательное число.

Пример: исходный код программы

```
1  /* fprintf example */
2  #include <stdio.h>
3
4  int main ()
5  {
6      FILE * pFile;
7      int n;
8      char name [100];
9
10     pFile = fopen ("myfile.txt", "w");
11     for (n=0 ; n<3 ; n++)
12     {
13         puts ("please, enter a name: ");
14         gets (name);
15         fprintf (pFile, "Name %d [%-10.10s]n", n, name);
16     }
17     fclose (pFile);
18
19     return 0;
20 }
```

Пример работы программы

В этом примере программа 3 раза запрашивает имя пользователя, а затем записывает их в файл `name.txt` каждый из них в соответствии с фиксированной длиной (всего 19 символов + строка).

Используется два тега форматирования:

`%d` : десятичное число

`%-10.10s` : выровненное влево (-), не менее десяти символов (10), максимум десять символов (0,10), String (s).

Предполагая, что мы ввели Дин, Сэм и Кастиэль. Итак `name.txt` должен содержать 3 имя:

```
Имя 0 [Дин ]
Имя 1 [Сэм ]
Имя 2 [Кастиэль]
```

Дополнительные примеры форматирования смотреть в описании функции `printf`.

Обсудить на форуме (/topics/)

Автор: admin (/forums/users/admin/)

Дата: 10.09.2012

Поделиться:

Похожие статьи:

1. Структура lconv (<http://cppstudio.com/post/1052/>)
2. Функции в C++ (<http://cppstudio.com/post/396/>)
3. Кириллица в консоли (<http://cppstudio.com/post/435/>)
4. Форматированный ввод/вывод в C++ (<http://cppstudio.com/post/319/>)
5. Аргументы функций по умолчанию (<http://cppstudio.com/post/409/>)

Комментарии

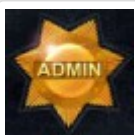


Benn

13.04.2013 (/post/1270/comment-page-1/#comment-185)

На лицо любитель «Сверхъестественного»)

Войдите, чтобы ответить (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F1270%2F)



admin (<http://cppstudio.com>)

14.04.2013 (/post/1270/comment-page-1/#comment-190)









На лицо любитель «Бойцовского клуба»)

Войдите, чтобы ответить (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F1270%2F)

Оставить комментарий

Вы должны войти (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F1270%2F), чтобы оставить комментарий.

Translation

 (/post/1270/)Русский (/post/1270/)
 (/uk/post/1270/)Українська (/uk/post/1270/)
 (/en/post/1270/)English (/en/post/1270/)
 (/de/post/1270/)Deutsch (/de/post/1270/)
 (/be/post/1270/)Беларуская (/be/post/1270/)
 (/kk/post/1270/)Қазақ тілі (/kk/post/1270/)
 (/uz/post/1270/)O'zbek tili (/uz/post/1270/)
 (/tr/post/1270/)Türkçe (/tr/post/1270/)

Новое

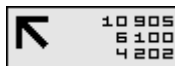
- Особенности Qt: слоты и сигналы, описание QObject и QApplication, виды окон и т.д.
(<http://cppstudio.com/post/11167/>)
- Первая программа на Qt:
(<http://cppstudio.com/post/11127/>)
- Введение – графическая библиотека Qt
(<http://cppstudio.com/post/11097/>)
- Наследование классов
(<http://cppstudio.com/post/10103/>)
- Перегрузка операторов в C++ (часть 2)
(<http://cppstudio.com/post/10058/>)

Популярное

Sorry. No data so far.



(<https://plus.google.com/u/0/106109650739084338784>)



(<http://www.liveinternet.ru/click>)



(<http://orphus.ru>)