

🏠 (<http://cppstudio.com>) / Язык программирования C++ (<http://cppstudio.com/cat/274/>)
/ Функции, локальные и глобальные переменные, рекурсия (<http://cppstudio.com/cat/274/280/>)
/ Прототипы функций в C++

Прототипы функций в C++



Оценка: **4,75** (голосов: **4**)

Чтобы проголосовать, вы должны зарегистрироваться.

Допустим, что в некотором месте в вашем коде происходит вызов определенной функции. Зададимся вопросом, при каких условиях при компилировании этого участка кода компилятор не выдаст ошибки. На этот вопрос существует простой ответ. Где-то в том же файле, в котором осуществляется вызов функции, перед операцией, в которой осуществляется вызов, должны присутствовать прототип или описание функции. Кроме того, аргументы и тип возвращаемого значения в вызове должны соответствовать аргументам и типу возвращаемого значения в прототипе и в описании функции.

Итак, что же такое прототип функции? Прототип имеет следующий вид.

```
1 // прототип функции
2 void function(int arg1, double arg2); // в конце определения всегда ставится точка
```

Как видно в прототипе указываются по порядку тип возвращаемого значения (в данном примере `void`), название функции (в данном случае `function`) и список параметров в скобках. Объявление прототипа должно заканчиваться точкой с запятой.

Для чего нужен прототип функции? Прототип и описание функции используются компилятором для того, чтобы вызов функции происходил правильным образом. Для этого компилятор сначала смотрит имя вызываемой функции и ищет в файле прототип или описание этой функции. Если найден прототип или описание, то проверяются аргументы, передаваемые функции в вызове, и использование возвращаемого значения.

Для чего нужен именно прототип? Почему нельзя ограничиться использованием одного описания функции? Прототип стал необходим после того, как стандарты языка C изменились таким образом, что перед вызовом функции в файле необходимо каким либо образом ее описать. Проблема состоит в том, что имя функции имеет глобальную область видимости (если ее описание находится вне всяких локальных областей). Допустим, что описание функции находится в отдельном исходном файле. Также допустим, что необходимо осуществить вызов этой же функции в нескольких других исходных файлах. Если нет прототипа, то в каждый такой исходный файл необходимо включить полное описание функции. Компилятор будет интерпретировать это как переопределение. Если же мы используем прототип, то мы можем включать этот прототип в столько исходных файлов, сколько нам необходимо.

Как лучше всего использовать прототип функции. Лучше всего описание функции включить в отдельный исходный файл. После этого надо скомпилировать этот файл и получить объектный файл. Прототип следует поместить в заголовочный файл и включать его директивой `#include` в те исходные файлы, в которых присутствует вызов функции.

Практика



Обсудить на форуме (/topics/)

Автор: admin (/forums/users/admin/)

Дата: 25.08.2012

Поделиться:

←Функции в C++ (<http://cppstudio.com/post/396/>)

Перегрузка функций в C++ (<http://cppstudio.com/post/406/>)









Похожие статьи:

1. Классы в C++ (<http://cppstudio.com/post/439/>)
2. Рекурсия в C++ (<http://cppstudio.com/post/418/>)
3. Функции в C++ (<http://cppstudio.com/post/396/>)
4. Шаблоны функций в C++ (<http://cppstudio.com/post/5165/>)
5. Функции в языке программирования C (<http://cppstudio.com/post/6471/>)

Оставить комментарий

Вы должны войти (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F404%2F), чтобы оставить комментарий.

Translation

 (/post/404/)Русский (/post/404/)
 (/uk/post/404/)Українська (/uk/post/404/)
 (/en/post/404/)English (/en/post/404/)
 (/de/post/404/)Deutsch (/de/post/404/)
 (/be/post/404/)Беларуская (/be/post/404/)
 (/kk/post/404/)Қазақ тілі (/kk/post/404/)
 (/uz/post/404/)O'zbek tili (/uz/post/404/)
 (/tr/post/404/)Türkçe (/tr/post/404/)

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/>)

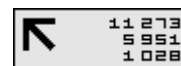
Популярное

Sorry. No data so far.

© 2022 CppStudio – Программирование для начинающих на C++



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



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



(<http://orphus.ru>)