Class implementation file

From Wikipedia, the free encyclopedia

In object-oriented programming, a **class implementation file** is often used to contain the implementation code for the method(s) of a class. This file is also referred to as a source file. Programming languages like C and C++ make use of these implementation files so as to separate the interface and implementation of these methods. [1]

Contents

Motivation

Using this structure, a class definition file containing the declaration of the class and its members is also created. If the class definition has been included and the implementation file for its methods is available, the user can instantiate an object of the class. The purpose of this structure is to keep the implementation code hidden, but allow the user to view the design. [2] [3]

Users make use of the public interface of an object so as to make creating objects as simple as possible, ensuring that client code does not distract the user with unnecessary details of the class's implementation. [4] This allows the user the information needed to use the class effectively, but prevents him or her from damaging the compiled code. [5]

The structure of a class implementation file

An implementation file is used in C++ programming when creating a class definition to split the interface from the implementation. The header file would declare all the member functions (methods) and data methods (fields) that the class has. [6] [7] [8]

The implementation file will contain the actual definition or source code of the methods declared in the header file. This file can start with a header block, which provides comments that describe the purpose of the defined class and any details about the creation of the actual file, such as the author of the file and date the file was created. [9] It can also include any libraries from the C++ Standard Library that will be used by any of the declared methods in the file. The **class implementation file** will usually have a line to include the associated header file (See examples below).

Example in C++

An example would be having a class called "ExampleClass". The header file of this C++ file would be named "ExampleClass.hpp" (though some may prefer "ExampleClass.h", despite it being a C convention) and the implementation file would be "ExampleClass.cpp". [10] [11]

An example of the structure of **ExampleClass.cpp** would look like this:

```
#include "ExampleClass.hpp"

ExampleClass::ExampleClass(): name(value std::string) {
   ...
}
void ExampleClass::addSomething(int k) {
   ...
}
```

In this example the coding for the functions has been omitted, but the methods "ExampleClass(): name(value std::string)" and "addSomething(int k)" would have been declared in **ExampleClass.hpp** like this: [12]

```
#include <string>
#include <iostream>

using namespace std;

class ExampleClass {
  public:
    ExampleClass();                    // Constructor
    void addSomething(int k);

  private:
   std::string name;
};
```

Example in Objective-C

Another example of how a class implementation file would be structured can be seen with Objective-C, which is used in iOS programming. [13] This example will use "ExampleClass". A notable difference between C++ and Objective-C when making use of these implementation files is the extensions used at the end of the files. In C++ it will be **.cpp** [14] and in Objective-C it will be **.m** [15] , but both will use the same **.h** extension for their header file(s) [16] [17] as shown in the example below.

This is an example of **ExampleClass.h** in Objective-C:

```
#import <UIKit/UIKit.h>

@interface ExampleClass : NSObject {
    // instance variable declarations go here
}
- (NSString*) name;
@end
```

This is an example of the **class's implementation file Exampleclass.m** in Objective-C:

```
#import "ExampleClass.h"

@implementation ExampleClass
- (NSString*) name {
    return @"…";
}
@end
```

See also

- C++ classes
- Header file
- Source code
- C++ Standard Library
- Objective-C

References

1. Alan Griffiths (2005). "Separating Interface and Implementation in C++". http://accu.org/index.php/journals/. ACCU. Retrieved 2013-05-07.
2. Alan Griffiths (2005). "Separating Interface and Implementation in C++". http://accu.org/index.php/journals/. ACCU. Retrieved 2013-05-07.
3. Neuberg, Matt (26 May 2011). "Chapter 4.3 Header File and Implementation File". *Programming iOS 4*. O'Reilly Media, Inc. ISBN 978-1-4493-8843-0.
4. Alan Griffiths (2005). "Separating Interface and Implementation in C++". http://accu.org/index.php/journals/. ACCU. Retrieved 2013-05-07.
5. "C++ Dos and Don'ts". http://www.chromium.org/developers/coding-style/cpp-dos-and-donts. The Chromium Projects. Retrieved 2013-05-07.
6. "Introduction to C++ Classes". http://pages.cs.wisc.edu/~hasti/cs368/CppTutorial/NOTES/. Retrieved 2013-05-07.
7. Alan Griffiths (2005). "Separating Interface and Implementation in C++". http://accu.org/index.php/journals/. ACCU. Retrieved 2013-05-07.
8. Febil Chacko Thanikal (2009). "How to define a template class in a .h file and implement it in a .cpp file". http://www.codeproject.com/Articles/48575/How-to-define-a-template-class-in-a-h-file-and-imp. Code Project. Retrieved 2013-05-07.
9. "The implementation file in C++ Programming". http://www.itechtalk.com/. ITechTalk. Retrieved 2013-05-07.
10. "Introduction to C++ Classes". http://pages.cs.wisc.edu/~hasti/cs368/CppTutorial/NOTES/CLASSES-INTRO.html. Retrieved 2013-05-07.
11. Neuberg, Matt (26 May 2011). "Chapter 4.3 Header File and Implementation File". *Programming iOS 4*. O'Reilly Media, Inc. ISBN 978-1-4493-8843-0.
12. "Introduction to C++ Classes". http://pages.cs.wisc.edu/~hasti/cs368/CppTutorial/NOTES/CLASSES-INTRO.html. Retrieved 2013-05-07.
13. Neuberg, Matt (26 May 2011). "Chapter 4.3 Header File and Implementation File". *Programming iOS 4*. O'Reilly Media, Inc. ISBN 978-1-4493-8843-0.
14. "Introduction to C++ Classes". http://pages.cs.wisc.edu/~hasti/cs368/CppTutorial/NOTES/. Retrieved 2013-05-07.
15. Neuberg, Matt (26 May 2011). "Chapter 4.3 Header File and Implementation File". *Programming iOS 4*. O'Reilly Media, Inc. ISBN 978-1-4493-8843-0.
16. "Introduction to C++ Classes". http://pages.cs.wisc.edu/~hasti/cs368/CppTutorial/NOTES/. Retrieved 2013-05-07.
17. Neuberg, Matt (26 May 2011). "Chapter 4.3 Header File and Implementation File". *Programming iOS 4*. O'Reilly Media, Inc. ISBN 978-1-4493-8843-0.

External links

- Header File and Implementation File (http://my.safaribooksonline.com/book/programming/iphone/9781449397302/objective-c-classes/_header_file_and_implementation_file)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Class_implementation_file&oldid=763448377"

Categories: Class (computer programming) | Object-oriented programming languages | C++ | Articles with example Objective-C code

- This page was last modified on 3 February 2017, at 09:15.
-