**异常处理（C# 编程指南）**

C# 程序员可使用 [try](https://msdn.microsoft.com/zh-cn/library/0yd65esw.aspx) 块对可能受异常影响的代码进行分区。关联的 [catch](https://msdn.microsoft.com/zh-cn/library/0yd65esw.aspx) 块用于处理任何结果异常。一个包含代码的 [finally](https://msdn.microsoft.com/zh-cn/library/zwc8s4fz.aspx) 块，无论 **try** 块中是否引发异常（例如，释放在 **try** 块中分配的资源），这些代码都会运行。一个 **try** 块需要一个或多个关联的 **catch** 块或一个 **finally** 块，或两者。

以下示例给出了一个 **try-catch** 语句，一个 **try-finally** 语句，和一个 **try-catch-finally** 语句。

try

{

// Code to try goes here.

}

catch (SomeSpecificException ex)

{

// Code to handle the exception goes here.

// Only catch exceptions that you know how to handle.

// Never catch base class System.Exception without

// rethrowing it at the end of the catch block.

}

try

{

// Code to try goes here.

}

finally

{

// Code to execute after the try block goes here.

}

try

{

// Code to try goes here.

}

catch (SomeSpecificException ex)

{

// Code to handle the exception goes here.

}

finally

{

// Code to execute after the try (and possibly catch) blocks

// goes here.

}

不带有 **catch** 或 **finally** 块的 **try** 块将导致编译器错误。

**Catch 块**

**catch** 块可以指定要捕捉的异常的该类型。类型规范称为“异常筛选器”。异常类型应从 [Exception](https://msdn.microsoft.com/zh-cn/library/system.exception.aspx) 派生出来。一般而言，不会将 [Exception](https://msdn.microsoft.com/zh-cn/library/system.exception.aspx) 指定为异常筛选器，除非您了解如何处理 **try** 块中可能引发的所有异常，或者您在 **catch** 块中包括了 [throw](https://msdn.microsoft.com/zh-cn/library/1ah5wsex.aspx) 语句。

具有不同异常筛选器的多个 **catch** 块可以串联在一起。多个 **catch** 数据块的计算顺序是在代码中从顶部到底部，但是，对于所引发的每个异常，都只执行一个 **catch** 数据块。与指定的准确类型或其基类最为匹配的第一个 **catch** 块被执行。如果 **catch** 块没有指定匹配异常筛选器，则 **catch** 块就不具有选定的筛选器（如果语句有的话）。需要将带有最具体的（即派生程度最高的）异常类的 **catch** 块放在最前面。

当下列条件为真时，应该捕捉异常：

* 对引发异常的原因有具体的了解，并可实现特定的恢复，例如，在捕获 [FileNotFoundException](https://msdn.microsoft.com/zh-cn/library/system.io.filenotfoundexception.aspx) 对象时提示用户输入新的文件名。
* 可以新建一个更具体的异常并引发该异常。
* int GetInt(int[] array, int index)
* {
* try
* {
* return array[index];
* }
* catch(System.IndexOutOfRangeException e)
* {
* throw new System.ArgumentOutOfRangeException(
* "Parameter index is out of range.");
* }
* }
* 希望在将异常传递出去进行额外处理前部分地处理异常。在下面的示例中，**catch** 块用于在再次引发异常之前，向错误日志添加条目。
* try
* {
* // Try to access a resource.
* }
* catch (System.UnauthorizedAccessException e)
* {
* // Call a custom error logging procedure.
* LogError(e);
* // Re-throw the error.
* throw;
* }

**Finally 块**

可以使用 **finally** 块清理在 **try** 块中执行的操作。如果存在，**finally** 块将在最后执行，在 **try** 块和任何匹配 **catch** 的块之后执行。不管是否引发异常或者是否找到与异常类型匹配的 **catch** 块，**finally** 始终运行。

可以使用 **finally** 块释放资源（如文件流、数据库连接和图形句柄），而不用等待由运行时中的垃圾回收器来完成对象。有关更多信息，请参见[using 语句（C# 参考）](https://msdn.microsoft.com/zh-cn/library/yh598w02.aspx)。

在下面的示例中，使用 **finally** 块关闭在 **try** 块中打开的文件。注意，在关闭文件之前要检查该文件句柄的状态。如果 **try** 块无法打开文件，则文件句柄仍具有值 **null**，并且 **finally** 块不会尝试关闭它。或者，如果在 **try** 块中成功打开该文件，则 **finally** 块将关闭打开的文件。

System.IO.FileStream file = null;

System.IO.FileInfo fileinfo = new System.IO.FileInfo("C:\\file.txt");

try

{

file = fileinfo.OpenWrite();

file.WriteByte(0xF);

}

finally

{

// Check for null because OpenWrite might have failed.

if (file != null)

{

file.Close();

}

}