

第八章

异常





一、异常语法



二、.Net框架中的异常类



三、处理异常



四、创建用户自定义异常



五、Throw



一、异常语法



异常是在程序执行期间出现的问题。例外是对程序运行时出现的异常情况的响应，例如尝试除以零。

异常提供了一种将控制从程序的一个部分转移到另一个部分的方法。VB.Net异常处理建立在四个关键字：Try，Catch，Finally和Throw。

Try: A Try block identifies a block of code for which particular exceptions will be activated. It's followed by one or more Catch blocks. Try块标识将激活特定异常的代码块。它后面是一个或多个Catch块。

Catch: A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The Catch keyword indicates the catching of an exception. 程序捕获异常，并在程序中要处理问题的位置使用异常处理程序。Catch关键字表示捕获异常。

Finally: The Finally block is used to execute a given set of statements, whether an exception is thrown or not thrown. For example, if you open a file, it must be closed whether an exception is raised or not. 最后：Finally块用于执行给定的一组语句，无论是抛出还是不抛出异常。例如，如果打开一个文件，那么无论是否引发异常，都必须关闭该文件。

Throw: A program throws an exception when a problem shows up. This is done using a Throw keyword. 当出现问题时，程序抛出异常。这是使用Throw关键字完成的。

一、异常语法



语法

假设块将引发异常，则方法使用Try和Catch关键字的组合捕获异常。 Try / Catch块放置在可能生成异常的代码周围。 Try / Catch块中的代码称为受保护代码，使用Try / Catch的语法如下所示：

```
Try
    [ tryStatements ]
    [ Exit Try ]
[ Catch [ exception [ As type ] ] [ When expression ]
    [ catchStatements ]
    [ Exit Try ] ]
[ Catch ... ]
[ Finally
    [ finallyStatements ] ]
End Try
```

您可以列出多个catch语句以捕获不同类型的异常，以防您的try块在不同情况下引发多个异常。

二、.Net框架中的异常类



在.Net框架中，异常由类表示。 .Net Framework中的异常类主要直接或间接从System.Exception类派生。 从System.Exception类派生的一些异常类是System.ApplicationException和System.SystemException类。

System.ApplicationException类支持由应用程序生成的异常。 所以程序员定义的异常应该从这个类派生。

System.SystemException类是所有预定义系统异常的基类。

下表提供了从Sytem.SystemException类派生的一些预定义异常类：

二、.Net框架中的异常类



异常类	描述
System.IO.IOException	Handles I/O errors. 处理I / O错误。
System.IndexOutOfRangeException	Handles errors generated when a method refers to an array index out of range. 当处理的方法是指一个数组索引超出范围产生的错误。
System.ArrayTypeMismatchException	Handles errors generated when type is mismatched with the array type 处理类型与数组类型不匹配时生成的错误。.
System.NullReferenceException	Handles errors generated from deferencing a null object. 处理从取消引用空对象生成的错误。
System.DivideByZeroException	Handles errors generated from dividing a dividend with zero. 处理将股利除以零所产生的错误。
System.InvalidCastException	Handles errors generated during typecasting. 处理类型转换期间生成的错误。
为System.OutOfMemoryException	Handles errors generated from insufficient free memory. 处理来自可用内存不足产生的错误。
System.StackOverflowException	Handles errors generated from stack overflow. 处理来自堆栈溢出产生的错误。

三、 处理异常



VB.Net提供了一个结构化的解决方案，以try和catch块的形式处理异常处理问题。使用这些块，核心程序语句与错误处理语句分离。这些错误处理块使用Try，Catch和Finally关键字实现。 以下是在零除条件时抛出异常的示例：

三、处理异常



```
Sub division(ByVal num1 As Integer, ByVal num2 As Integer)
    Dim result As Integer
    Try
        result = num1 / num2
    Catch e As DivideByZeroException
        Console.WriteLine("Exception caught: {0}", e)
    Finally
        Console.WriteLine("Result: {0}", result)
    End Try
End Sub
Sub Main()
    division(25, 0)
    Console.ReadKey()
End Sub
```


四、创建用户自定义异常



您还可以定义自己的异常。用户定义的异常类派生自ApplicationException类。以下示例演示了这一点：

```
Public Class TempIsZeroException : Inherits ApplicationException
    Public Sub New(ByVal message As String)
        MyBase.New(message)
    End Sub
End Class
Public Class Temperature
    Dim temperature As Integer = 0
    Sub showTemp()
        If (temperature = 0) Then
            Throw (New TempIsZeroException("Zero Temperature found"))
        Else
            Console.WriteLine("Temperature: {0}", temperature)
        End If
    End Sub
End Class
```

五、Throw



如果它是直接或间接从System.Exception类派生的，你可以抛出一个对象。

你可以在catch块中使用throw语句来抛出当前对象：

Throw [expression]

```
' Throw
Try
    Throw New ApplicationException("A custom exception _
is being thrown here...")
Catch e As Exception
    Console.WriteLine(e.Message)
Finally
    Console.WriteLine("Now inside the Finally Block")
End Try
```