C++异常处理机制（throw、try、catch、finally） - 知道了呀~ - 博客园

知道了呀~ 粉丝 - 82 关注 - 21

一、什么是异常处理

一句话：异常处理就是处理程序中的错误。

程序运行时常会碰到一些异常情况，例如：

1、做除法的时候除数为 0；

2、用户输入年龄时输入了一个负数；

3、用 new 运算符动态分配空间时，空间不够导致无法分配；

4、访问数组元素时，下标越界；打开文件读取时，文件不存在。

这些异常情况，如果不能发现并加以处理，很可能会导致程序崩溃。

二、异常处理机制

1、当发生异常，程序无法沿着正常的顺序执行下去的时候，立即结束程序可能并不妥当。我们需要给程序提供另外一条可以安全退出的路径，

在结束前做一些必要的工作，如将内存中的数据写入文件、关闭打开的文件、释放动态分配的内存空间等。

2、当发生异常的时候，程序马上处理可能并不妥当（一个异常有多种处理方法，或者自己无法处理异常），需要将这个异常抛出给他的上级（直接调用者），

由上级决定如何处理。或者是自己不处理再转交给它的上级去处理，一直可以转交到最外层的main()函数

3、另外，异常的分散处理不利于代码的维护，尤其是对于在不同地方发生的同一种异常，都要编写相同的处理代码也是一种不必要的重复和冗余。

如果能在发生各种异常时让程序都执行到同一个地方，这个地方能够对异常进行集中处理，则程序就会更容易编写、维护。

在引入异常处理机制之前，异常的处理方式有两种方法

1、使用整型的返回值标识错误；

2、使用errno宏（可以简单的理解为一个全局整型变量）去记录错误。当然C++中仍然是可以用这两种方法的。

这两种方法最大的缺陷就是会出现**不一致**问题。例如有些函数返回1表示成功，返回0表示出错；而有些函数返回0表示成功，返回非0表示出错。

还有一个缺点就是函数的返回值只有一个，你通过函数的返回值表示错误代码，那么函数就不能返回其他的值。

鉴于上述原因，C++引入了异常处理机制

异常处理流程

C++ 异常处理涉及到三个关键字：**try、catch、throw**。

1、throw: 当问题出现时，程序会抛出一个异常。这是通过使用 **throw** 关键字来完成的。

2、try: try 块中的代码标识将被激活的特定异常。它后面通常跟着一个或多个 catch 块。

3、catch: 在您想要处理问题的地方，通过异常处理程序捕获异常。**catch** 关键字用于捕获异常。

4、finally：关键字finally放在catch之后，如果异常没有被catch捕获，会使用关键字去清理释放资源

如果有一个块抛出一个异常，捕获异常的方法会使用 **try** 和 **catch** 关键字。try 块中放置可能抛出异常的代码（判断异常的类型），try 块中的代码被称为保护代码。

catch后面对应每个异常的处理方法。

以除法除0举例。代码如下所示：

```
#include <iostream>
using namespace std;

double division(int a, int b)
{
  if (b == 0)
  {
    throw "Division by zero condition!";
  }
  return (a / b);
}

int main()
{
  int x = 50;
  int y = 0;
  double z = 0;
  //try\catch的使用和switch\case的使用类似
  try {
    z = division(x, y);
    cout << z << endl;
  }
  catch (const char* msg) {
    cerr << msg << endl;
  }
    //finally{}
  return 0;
}
```

三、几个概念

1、栈展开

栈展开指的是：当异常抛出后，匹配catch的过程。

抛出异常时，将暂停当前函数的执行，开始查找匹配的catch子句。沿着**函数的嵌套调用链向上查找**，直到找到一个匹配的catch子句，或者找不到匹配的catch子句。

栈展开的时候，会通过析构函数或者是delete销毁局部对象(从开始匹配位置到确认匹配这一段中间位置的资源会被释放)

2、析构函数应该从不抛出异常。

如果析构函数中出现异常，那么就应该在析构函数**内部**将这个异常进行处理，而不是将异常抛出去。

为什么不应该？抛出异常的就是栈展开的过程，而栈展开会调用析构函数销毁局部对象，这样多次调用析构函数会导致程序崩溃(内存泄漏)

3、构造函数可以抛出异常

当构造函数内出现异常，可以选择将异常抛出，在栈展开的过程调用析构函数释放已申请的内存，也可以在内部将异常处理，手动调用delete释放

4、catch捕获所有异常

语法：在catch语句中，使用**三个点（...）**。即写成：catch (...) 这里三个点是“通配符”，类似 可变长形式参数。