面向对象程序设计

C++ 的异常处理

2020 年 春

耿楠

计算机科学系
信息工程学院

西北农林科技大学
NORTHWEST A&F UNIVERSITY
中国 · 杨凌

- 错误：语法错误，逻辑错误和运行错误。
- 常见异常错误：数组下标越界、运算溢出、除 0、动态分配内存失败和文件读写失败。
- 异常处理机制：`try-throw-catch`

- 能够抛出异常的语句被包围在 try 块中。

- try 块以关键字 try 开始，后面是花括号括起来的语句序列。try 块之后紧跟一组处理代码，称为 catch 子句。try 块将语句分成组，并将它们与处理这些语句可能抛出的异常的语句相关联。

- C++ 的异常处理代码是 catch 子句。当一个异常被 try 块中的语句抛出时，系统在 try 块后的 catch 子句列表中查找能够处理该异常的 catch 子句。

一元二次方程的根

```
struct Solution
{
   double x1, x2;
   Solution(double x1 = 0, double x2 = 0)
   {
      this->x1 = x1;
      this->x2 = x2;
   }
   friend ostream &operator<<(ostream &out, Solution s)
   {
      if(s.x1 == s.x2)
         out << "x1=x2=" << s.x1;
      else
         out << "x1=" << s.x1 << "," << "x2=" << s.x2;
      return out;
   }
};

Solution FindRoots(double a, double b, double c)
{
   double x1, x2;

   if(abs(a) < 1.0e-8)
   {
      if(abs(b) < 1.0e-8)
         throw a;
      else
         return Solution(-c / b, -c / b);
   }
   if((b * b - 4 * a * c) < 0) throw (b * b - 4 * a * c);
   x1 = (-b + sqrt(b * b - 4 * a * c)) / (2 * a);
   x2 = (-b - sqrt(b * b - 4 * a * c)) / (2 * a);
   return Solution(x1, x2);
}
```

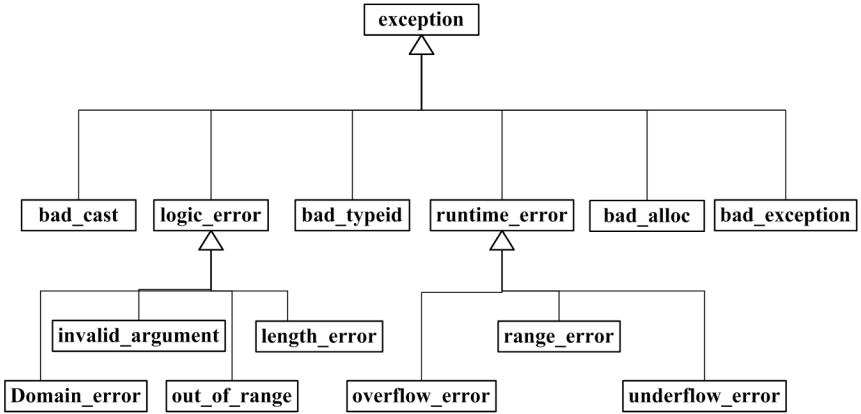
- 一元二次方程的根

```
#include <iostream>
#include <cmath>

using namespace std;

int main()
{
   try
   {
      cout << FindRoots(0, 2, 2) << endl;
      cout << FindRoots(1, 2, 1) << endl;
      cout << FindRoots(1, 4, 1) << endl;
      cout << FindRoots(0, 0, 1) << endl;
      cout << FindRoots(-1, 9, 3) << endl;
   }

   catch(double x)
   {
      if(abs(x) < 1.0e-8) cout << "Deviding zero!" << endl;
      if(x < 0) cout << "The roots are complex!" << endl;
   }

   cout << "Continue..." << endl;
   return 0;
}
```

文件异常

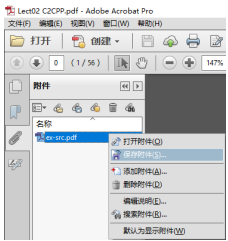
```cpp
#include <iostream>
#include <fstream>
using namespace std;

int main ()
{
   ifstream file;
   file.exceptions ( ifstream::failbit | ifstream::badbit );
   try
   {
      file.open ("test.txt");
      while (!file.eof()) file.get();
   }
   catch (ifstream::failure e)
   {
      cout << "Exception opening/reading file";
   }

   file.close();

   return 0;
}
```

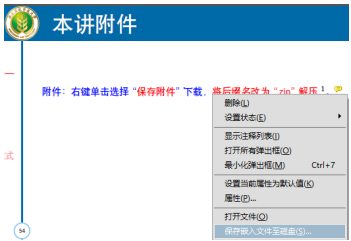
系统预定义异常

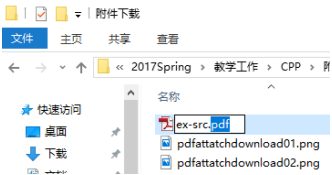


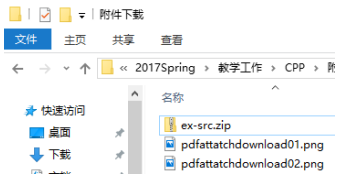
- 集中处理容易出问题的代码。
- 异常只是错误处理技术的一种，程序中还应该使用其他错误处理技术，如断言、返回错误代码等。
- 异常使程序的复杂性增加，可以参考一些指导原则，慎用异常机制。

附件：右键单击该链接，选择“保存附件”下载，将后缀名改为“.zip”解压[1][2]。









[1]请退出全屏模式后点击该链接。

[2]以 Adobe Acrobat Reader 为例。

本讲结束，谢谢!
欢迎多提宝贵意见和建议

西北农林科技大学
NORTHWEST A&F UNIVERSITY
中国 · 杨凌