

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

异常处理的方法

异常处理方法：三种语句的一般形式与用法

❏ throw语句

throw 表达式;

❏ try-catch的结构为

try

{

被检查的语句

}

catch(异常信息类型 [变量名])

{

进行异常处理的语句

}

catch(...)

{

... ..

}

❏ 被检测的函数必须放在try块中，否则不起作用。

❏ try块和catch块作为一个整体出现。

❏ try和catch块中必须用花括号括起来，即使花括号内只有一个语句。

❏ 一个try-catch结构中只能有一个try块，但却可以有多个catch块，以便与不同的异常信息匹配。

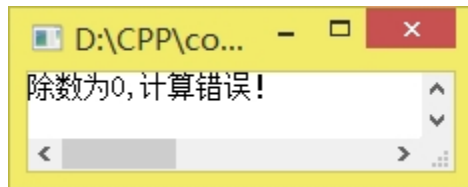
❏ catch后面的圆括号中，一般只写异常信息的类型名，如catch(double)

❏ 如果throw抛出的异常信息找不到与之匹配的catch块，那么系统就会调用一个系统函数terminate，使程序终止运行。

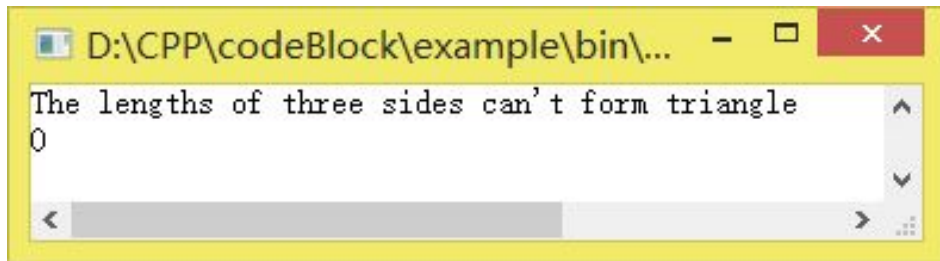
例：防止除数为0

```
#include <iostream>
using namespace std;
template <typename T>
T Div(T x,T y)
{
    if(y==0)
        throw y;//抛出异常
    return x/y;
}
```

```
int main()
{
    int x=5,y=0;
    double x1=5.5,y1=0.0;
    try {
        //被检查的语句
        cout<<x<<"/"<<y<<"="<<Div(x,y)<<endl;
        cout<<x1<<"/"<<y1<<"="<<Div(x1,y1)<<endl;
    }
    catch(int) { //异常类型
        cout<<"除数为0,计算错误！"<<endl;//异常处理语句
    }
    catch(double) { //异常类型
        cout<<"除数为0.0,计算错误！"<<endl;//异常处理语句
    }
    return 0;
}
```



异常处理再例：求三角形周长

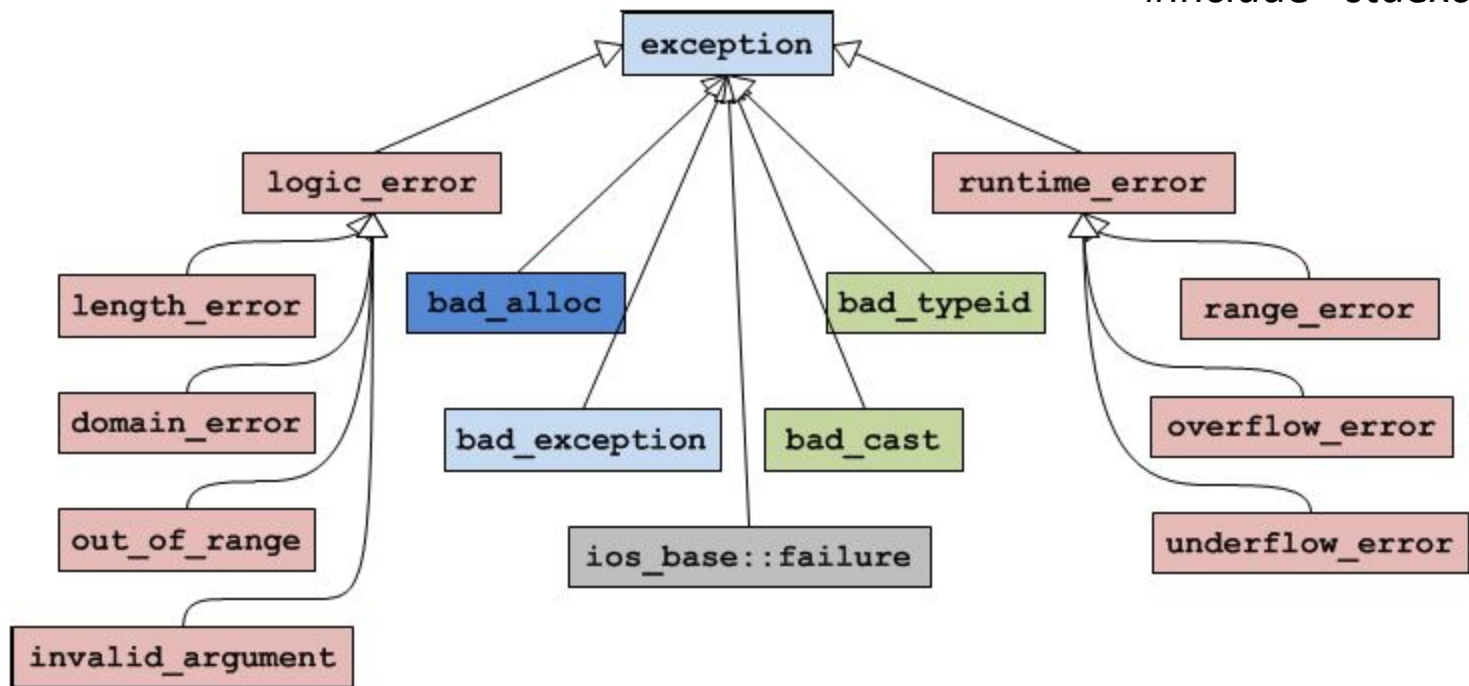


```
#include <iostream>
#include <stdexcept>
using namespace std;
int triangle(int a, int b, int c)
{
    if(a<0 || b<0 || c<0 || a+b<=c || a+c<=b || b+c<=a)
        throw runtime_error("The lengths of three sides can't form triangle");
    return a + b + c;
}
```

```
int main()
{
    int total = 0;
    try
    {
        total = triangle(3,4,7);
    }
    catch(const runtime_error& e)
    {
        cout<<e.what()<<endl;
    }
    cout<<total<<endl;
    return 0;
}
```

标准异常类

#include <stdexcept>

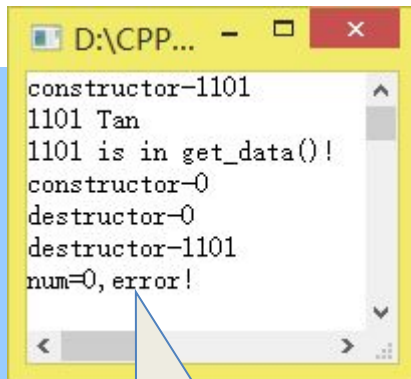


在异常处理中处理析构函数

```
class Student
{
public:
    Student(int n,string nam):num(n), name(nam)
    {
        cout<<"constructor-"<<n<<endl;
    }
    ~Student( )
    {
        cout<<"destructor-"<<num<<endl;
    }
    void get_data( );
private:
    int num;
    string name;
};
```

```
void Student::get_data( )
{
    if(num==0)
        throw num;
    else
        cout<<num<<" "<<name<<endl;
        cout<<num<<" is in get_data()!"<<endl;
}
```

善后工作只能指望catch块及析构函数。



```
constructor-1101
1101 Tan
1101 is in get_data()!
constructor-0
destructor-0
destructor-1101
num=0, error!
```

```
void fun( )
{
    Student stud1(1101,"Tan");
    stud1.get_data( );
    Student stud2(0,"Li");
    stud2.get_data( );
    cout<<"end of fun!"<<endl;
}
```

```
int main( )
{
    try
    {
        fun( );
    }
    catch(int n)
    {
        cout<<"num="<<n<<" ,error!"<<endl;
    }
    return 0;
}
```

在函数声明中也可以进行异常情况指定

- ❏ 为便于阅读程序，使用户在看程序时能够知道所用的函数是否会抛出异常信息以及异常信息可能的类型，C++允许在声明函数时列出可能抛出的异常类型

//triangle函数只能抛出double类型的异常信息

```
double triangle(double,double,double) throw(double);
```

//函数可以抛出int,double,float或char类型的异常信息

```
double triangle(double,double,double) throw(int,double,float,char);
```

//声明一个不能抛出异常的函数

```
double triangle(double,double,double) throw();
```

- ❏ 要指定异常类型，必须在声明和定义时都指定，且类型相同

在函数嵌套的情况下检测异常处理

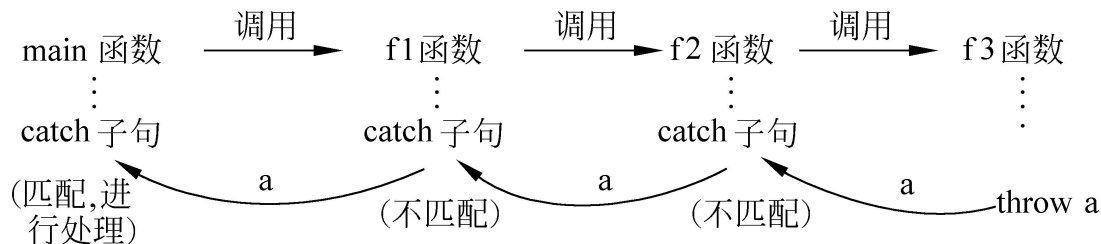
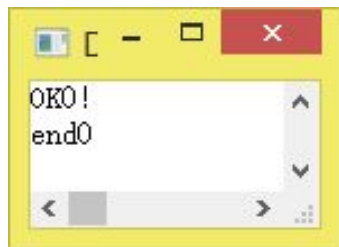
```
int main( )
{
    void f1( );
    try
    {
        f1(); //调用f1()
    }
    catch(double)
    {
        cout<<"OK0!"<<endl;
    }
    cout<<"end0"<<endl;
    return 0;
}
```

```
void f1( )
{
    void f2( );
    try
    {
        f2();
    }
    catch(char)
    {
        cout<<"OK1!";
    }
    cout<<"end1"<<endl;
}
```

```
void f2( )
{
    void f3( );
    try
    {
        f3();
    }
    catch(int)
    {
        cout<<"Ok2!"<<endl;
    }
    cout<<"end2"<<endl;
}
```

```
void f3( )
{
    double a=0;
    try
    {
        throw a;
    }
    catch(float)
    {
        cout<<"OK3!"<<endl;
    }
    cout<<"end3"<<endl;
}
```

改为int a = 0?



运筹帷幄，决胜千里



THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>