## 异常处理

#### 异常处理的介绍：

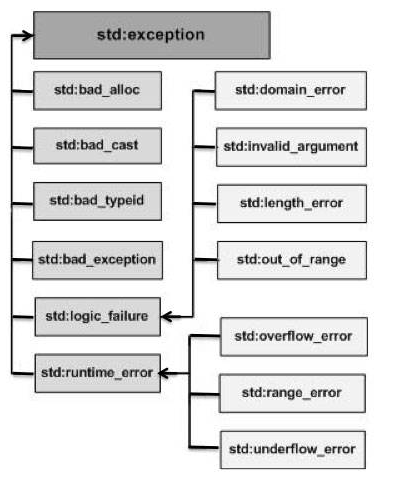
1. **异常是程序在执行期间产生的问题（编译期出现的错误在写代码时开发环境就有提示）**。C++的异常是指程序运行时发生的特殊情况。
2. 异常提供了一种转移程序控制权的方式。C++的异常处理涉及到三个关键字：try，catch，throw。
   1. throw：当问题出现时，程序会抛出一个异常。这是通过throw关键字来完成的。
   2. catch：在你想要处理问题的地方，通过异常处理程序捕获异常。catch关键字用于捕获异常。
   3. try：try块中的代码标识将被激活的特定异常。它后面通常跟着一个或多个catch块。

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块中放可能抛出异常的代码，try块中的代码被称为保护代码。常见的异常处理格式如图所示。

1. 抛出异常：throw语句可以在代码块的任何地方抛出异常，throw抛出的表达式的结果决定了抛出的异常的类型。

#### C++的标准异常

1. C++提供了一系列标准的异常，定义在头文件“<exception>”中，它们是以父子层次结构组织起来的，如下图所示。





1. 别看图很复杂，异常种类有很多，但经常使用的其实就几个。
   1. bad\_alloc错误，使用new分配内存失败就会抛出bad\_alloc错误。
   2. out\_of\_range错误，在使用at时，容器越界就会抛出这个错误，这也是“at”比“[]”更加优秀的原因。
   3. runtime\_error错误，运行时错误，只有在程序运行时才能检测到的错误。这是一个相对的概念，和logic\_error形成对比。logic\_error可以读代码读出来，runtime\_error就不行。

我们也经常将一些读代码无法判断的异常标识为runtime\_error。

* 1. ... 错误，可以接受任何错误，我们一般都会在catch最后加上“...”，这样就可以接受所有类型的异常了。

1. 自定义异常类型，其实需要自定义异常类型的情况真的非常少，这里就不介绍了，其实和标准异常也是一样的。