

语句

2021年10月28日 22:15

- 1、switch中的case标签必须是整型常量表达式。
- 2、switch中即使不准备在default标签下做任何工作，定义一个default标签也是有用的，即告诉程序的读者我们已经考虑到了默认的情况。
- 3、记得do while语句应该在括号包围起来的条件后面用一个分号表示语句结束。
- 4、do while语句条件部分的变量必须定义在循环体之外。
另外，因为对于do while来说先执行语句或者块，后判断条件，所以不允许在条件部分定义变量。
- 5、break语句的作用范围仅限于最近的循环或者switch。
continue语句终止最近的循环中的当前迭代并立即开始下一次迭代。
- 6、try语句块和异常处理：

异常处理机制为程序中异常检测和异常处理这两部分的协作提供支持。在 C++语言中，异常处理包括：

- **throw 表达式** (throw expression)，异常检测部分使用 throw 表达式来表示它遇到了无法处理的问题。我们说 throw 引发 (raise) 了异常。
- **try 语句块** (try block)，异常处理部分使用 try 语句块处理异常。try 语句块以关键字 try 开始，并以一个或多个 **catch 子句** (catch clause) 结束。try 语句块中代码抛出的异常通常会被某个 catch 子句处理。因为 catch 子句“处理”异常，所以它们也被称作**异常处理代码** (exception handler)。
- 一套**异常类** (exception class)，用于在 throw 表达式和相关的 catch 子句之间传递异常的具体信息。

throw表达式：

一种中断当前执行路径的表达式。throw表达式抛出一个异常并把控制权转移到能处理该异常的最近的catch子句。

程序的异常检测部分使用 throw 表达式引发一个异常。throw 表达式包含关键字 throw 和紧随其后的一个表达式，其中表达式的类型就是抛出的异常类型。throw 表达式后面通常紧跟一个分号，从而构成一条表达式语句。

try语句块：

跟在try关键字后面的块，以及一个或多个catch子句。如果try语句块的代码引发异常并且其中一个trycatch子句匹配该异常类型，则异常被该catch子句处理。否则，异常将由外围try语句块处理，或者程序终止。

try 语句块的通用语法形式是

```
try {  
    program-statements  
} catch (exception-declaration) {  
    handler-statements  
} catch (exception-declaration) {  
    handler-statements  
} // ...
```

try 语句块的一开始是关键字 try，随后紧跟着一个块，这个块就像大多数时候那样是花括号括起来的语句序列。

跟在 try 块之后的是一个或多个 catch 子句。catch 子句包括三部分：关键字 catch、括号内一个（可能未命名的）对象的声明（称作**异常声明**，exception declaration）以及一个块。当选中了某个 catch 子句处理异常之后，执行与之对应的块。catch 一旦完成，程序跳转到 try 语句块最后一个 catch 子句之后的那条语句继续执行。

try 语句块中的 *program-statements* 组成程序的正常逻辑，像其他任何块一样，*program-statements* 可以有包括声明在内的任意 C++语句。一如往常，try 语句块内声明的变量在块外部无法访问，特别是在 catch 子句内也无法访问。

标准异常：

C++标准库定义了一组类，用于报告标准库函数遇到的问题。这些异常类也可以在用户编写的程序中使用，它们分别定义在4个头文件中：

- exception 头文件定义了最通用的异常类 exception。它只报告异常的发生，不提供任何额外信息。
- stdexcept 头文件定义了几种常用的异常类，详细信息在表 5.1 中列出。
- new 头文件定义了 bad_alloc 异常类型，这种类型将在 12.1.2 节（第 407 页）详细介绍。
- type_info 头文件定义了 bad_cast 异常类型，这种类型将在 19.2 节（第 731 页）详细介绍。

表 5.1: <stdexcept>定义的异常类	
exception	最常见的问题
runtime_error	只有在运行时才能检测出的问题
range_error	运行时错误：生成的结果超出了有意义的值域范围
overflow_error	运行时错误：计算上溢
underflow_error	运行时错误：计算下溢
logic_error	程序逻辑错误
domain_error	逻辑错误：参数对应的结果值不存在
invalid_argument	逻辑错误：无效参数
length_error	逻辑错误：试图创建一个超出该类型最大长度的对象
out_of_range	逻辑错误：使用一个超出有效范围的值

习题：

练习5.14

练习 5.14: 编写一段程序，从标准输入中读取若干 string 对象并查找连续重复出现的单词。所谓连续重复出现的意思是：一个单词后面紧跟着这个单词本身。要求记录连续重复出现的最大次数以及对应的单词。如果这样的单词存在，输出重复出现的最大次数；如果不存在，输出一条信息说明任何单词都没有连续出现过。例如，如果输入是

```
how now now now brown cow cow
```

那么输出应该表明单词 now 连续出现了 3 次。