**1. 异常处理的基本流程**

* 当程序抛出异常时，会立即停止当前函数的执行，并开始查找匹配的 catch 块。
* 如果当前函数中没有匹配的 catch 块，程序会沿着调用栈向上查找，直到找到匹配的 catch 块或退出 main 函数。

**2. 栈展开（Stack Unwinding）**

* **栈展开**是指在抛出异常后，程序沿着调用栈向上查找匹配的 catch 块的过程。
* 在栈展开过程中，所有局部对象会被正确销毁（调用析构函数），确保资源被释放。

**3. 异常捕获**

* 如果找到匹配的 catch 块，程序会执行 catch 块中的代码。
* 执行完 catch 块后，程序会继续执行 try-catch 块之后的代码

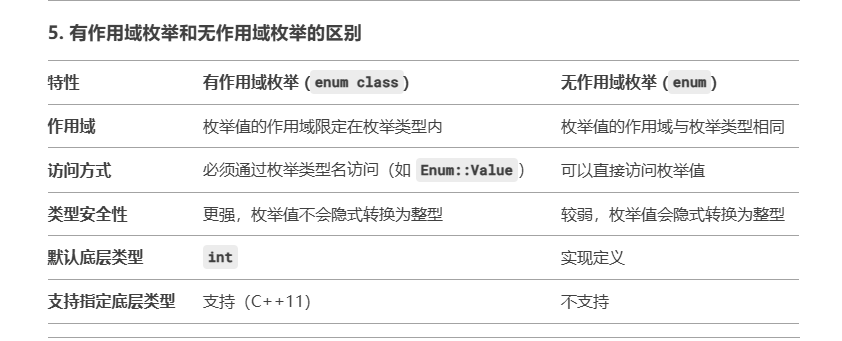
**4. 未捕获的异常**

* 如果异常没有被捕获（即没有找到匹配的 catch 块），程序会调用标准库函数 std::terminate。
* std::terminate 会终止程序的执行。
* 命名空间是C++中用于避免命名冲突和命名空间污染的重要机制。
* 通过将库的名称放在命名空间中，可以有效组织和管理全局名称。
* 命名空间的使用包括定义命名空间、使用作用域运算符、using 声明和 using 指令。

定义多个不相关类型的命名空间应使用单独的文件来表示命名空间定义的每个类型（或每个相关类型的集合）。3

未命名的命名空间替换文件静态 在引入命名空间之前，程序将名称声明为静态，以使它们成为文件的本地名称。 文件静态的使用继承自 C。在 C 中，声明为 static 的全局实体在声明它的文件之外是不可见的。 警告 C++ 标准不推荐使用文件静态声明。 应避免文件静态并使用未命名的名称空间。

* **多重继承**允许一个类从多个基类继承，但可能导致菱形继承问题。
* **虚继承**通过共享基类子对象解决菱形继承问题。
* 在使用多重继承和虚继承时，需要注意二义性和对象布局问题。



一个类可以在另一个类中定义。 这样的类就是嵌套类，也称为嵌套类型。 嵌套类最常用于定义实现类，例如我们在文本查询示例中使用的 QueryResult 类（第 12.3 节，第 484 页）。

C++ Primer，第五版 •

联合是一种特殊的类，它可以定义多个数据成员，但在任何时间点，只有一个成员可能具有值。 联合通常嵌套在另一个类类型中。

• 本地类，在函数内部定义。 本地类的所有成员都必须在类主体中定义。 局部类没有静态数据成员。 C++ 还支持一些固有的不可移植功能，包括位域和易失性（这使得与硬件的接口更容易）和链接指令（这使得与用其他语言编写的程序的接口更容易）。

**1. 枚举的基本概念**

* 枚举类型定义了一组命名的整型常量，称为**枚举值**（Enumerators）。
* 枚举类型定义了一个新的类型，类似于类。
* 枚举值默认从 0 开始，依次递增，但可以显式指定枚举值的整数值。

**1. 联合体的定义**

* 联合体使用 union 关键字定义。
* 联合体的所有成员共享同一块内存空间。
* 联合体的大小等于其最大成员的大小。

**1. 局部类的定义**

* 局部类在函数内部定义，作用域仅限于该函数。
* 局部类可以访问函数中的局部变量，但只能访问 static 局部变量。

**2. 局部类的特点**

* **作用域**：局部类的作用域仅限于定义它的函数内部。
* **访问权限**：
  + 局部类可以访问函数中的 static 局部变量。
  + 局部类不能访问函数中的非 static 局部变量。
* **成员函数**：局部类的成员函数必须在类内定义（不能分离定义）。
* **模板**：局部类不能定义模板。