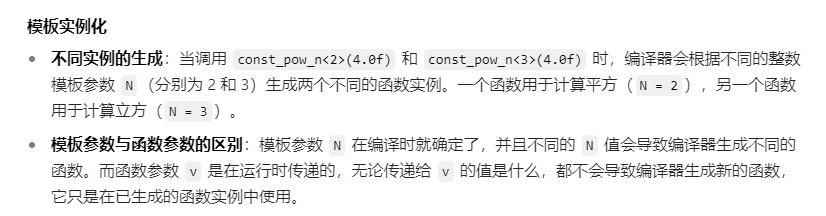
* **元编程**：元编程是一种更高层次的编程技术，它编写的代码的主要作用不是直接实现程序的功能，而是生成或转换其他代码。这些生成或转换后的代码才是最终会被编译成机器码的常规 C++ 代码。

C++ 提供了模板和常量表达式这两种机制，让开发者可以在 C++ 语言内部编写元程序。模板是一种泛型编程工具，它允许我们编写与类型无关的代码；常量表达式是指在编译时就能计算出结果的表达式。通过这两者，我们可以在编译阶段进行一些计算和代码生成的工作。

模版实例化是在编译的时候进行的。在 C++ 中，当编译器遇到模板的使用时，它会根据实际提供的模板参数来生成具体的函数或类的实例，这个过程就是模板实例化。



当模板参数的类型或值过多时，确实会导致代码膨胀，这是模板编程的一个显著缺点。

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

这对于理解元编程的工作原理很重要。模板代码生成非模板的 C++ 代码，然后这些代码会作为常规代码被执行。如果生成的 C++ 代码无法编译，错误将在编译时被捕获。

C++20 引入了一种新的缩写语法来编写函数模板，采用了与通用 lambda 表达式相同的风格。通过使用 auto 作为函数参数类型，实际上我们是在创建一个函数模板，而不是一个常规函数。

对于if语句，传入的条件可以在编译时确定的话，那么就会生成对应的表达式，而不是在运行的时候去执行

**使用常量表达式编程**

以 constexpr 关键字开头的表达式会告知编译器该表达式应在编译时求值