

Classes Enum e Objetos de Função

Classes enum

Nós já conhecemos a utilidade dos tipos enumerados, mas *não são recomendados* em C++, em alguns casos, permitem comparações que não fazem sentido.

C++11 define um novo conceito, a classe `enum` (também chamada de enumeração com escopo definido), que torna as enumerações com tipos e escopos fortes. Para fazer uma classe `enum`, usamos a palavra-chave `class` após a palavra-chave `enum`.

Com classes `enum`, o compilador não converterá mais implicitamente os valores do enumerador em inteiros.

A palavra-chave `class`, junto com a palavra-chave `static`, é uma das palavras-chave mais sobrecarregadas na linguagem C++ e pode ter significados diferentes dependendo do contexto. Embora as classes `enum` usem a palavra-chave `class`, elas não são consideradas “classes” no sentido tradicional do C++ .



Objetos de função

Um **functor**, ou function object (objeto de função), é um termo não oficial (sendo `function object` o termo oficial) usado por programadores em **C++**.

Um objeto de função é qualquer objeto para o qual o operador de chamada de função é definido. **C++** fornece muitos objetos de função integrados, bem como suporte para a criação e manipulação de novos objetos de função.

Um functor é diferente de uma função! Um functor é basicamente apenas uma classe que define o **`operator()`**. Isso permite criar objetos que "se parecem" com uma função. Um functor serve para quando precisamos passar um retorno como parâmetro de uma função.

