"A Besta continuou seus estudos com foco renovado, gerando grandes trabalhos de referência e contemplando novas realidades. A Besta produziu seus seguidores e acólitos para criar uma forma menor e renovada de si mesma e, através de meios travessos, a enviou pelo mundo" (O Livro da Mozilla, 6:27).

Enumerations

Paulo Ricardo Lisboa de Almeida

DSBD

UFPR

Tipos de Disciplina

Uma disciplina pode ser optativa ou mandatória.

Poderíamos criar uma variável `int` dentro de `Disciplina`, que recebe uma macro `OPT` ou `MAND`, como faríamos usando C.

```
#define OPT 1
#define MAND 2
```

Tipos de Disciplina

Uma disciplina pode ser optativa ou mandatória.

Poderíamos criar uma variável `int` dentro de `Disciplina`, que recebe uma macro `OPT` ou `MAND`, como faríamos usando C.

```
#define OPT 1
#define MAND 2
```

Mas #defines para gerar macros geralmente são uma **péssima ideia** em C++.

> **"You're not really going to define a macro, are you?"**
>
> https://google.github.io/styleguide/cppguide.html#Macro_Names
>
> https://google.github.io/styleguide/cppguide.html#Preprocessor_Macros

Scoped enums

Antes do C++11 a prática nesse caso era criar uma enum estilo C.

- São similares a constantes globais.
- Logo, são uma má ideia.

A partir do C++11, podemos (devemos) criar enums com escopo.

Crie um .hpp com a Enumeration

```
#ifndef ENUM_TIPO_DISCIPLINA_HPP
#define ENUM_TIPO_DISCIPLINA_HPP

namespace ufpr{
enum class EnumTipoDisciplina{
   OPTATIVA,
   MANDATORIA
};
}
#endif
```

Crie um .hpp com a Enumeration

```
#ifndef ENUM_TIPO_DISCIPLINA_HPP
#define ENUM_TIPO_DISCIPLINA_HPP

namespace ufpr{
enum class EnumTipoDisciplina{
   OPTATIVA,
   MANDATORIA
};
}
#endif
```

Por padrão, o tipo de cada item da enum é int.

O primeiro item é o 0, o segundo é o 1, ...

Resolução de Escopo

Use o operador de resolução de escopo :: para acessar o valor de uma das constantes da enum.

```cpp
#include <iostream>

#include "Disciplina.hpp"
#include "EnumTipoDisciplina.hpp"

using namespace ufpr;

int main(){
   Disciplina d1{"Programacao", nullptr, EnumTipoDisciplina::OPTATIVA};
   d1.setCargaHoraria(30);

   std::cout << d1.getNome() << '\n';

   return 0;
}
```

Nomenclatura

O nome da enum deve seguir as mesmas regras de nomenclatura da criação de classes.

Os valores da enum comumente ficam com todos caracteres em UPPER CASE, como em um #define.

Obs.: o Google não gosta dessa prática.

Parece que internamente eles usam muitas libs mal feitas (deles será?) que possuem #define, gerando colisões de nome nos sistemas.

https://google.github.io/styleguide/cppguide.html#Enumerator_Names

Começando a contagem de outro valor

Você pode modificar o comportamento padrão, e começar a contagem por outro valor.

Exemplo:

```
enum class Meses {
    JAN = 1,
    FEB,
    MAR,
    APR,
    MAY,
    JUN,
    JUL,
    AUG,
    SEP,
    OCT,
    NOV,
    DEC
};
```

Tipos

Você pode modificar o tipo padrão de enums.

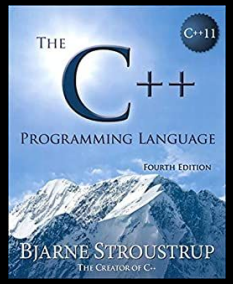
```
enum class Status : unsigned int {CONTINUE, WON, LOST};
```

Exercícios

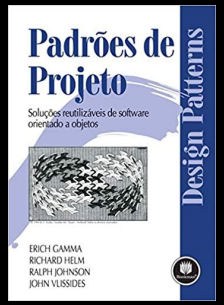
1. A classe disciplina não deve aceitar uma carga horária menor do que 30 horas para uma disciplina mandatória. Disciplinas optativas podem ter qualquer carga horária. Modifique esse comportamento na classe disciplina, lançando uma exceção caso um valor inválido seja passado.

Referências

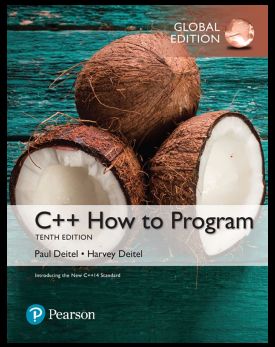
Bjarne Stroustrup. The C++ Programming Language. Addison-Wesley, 2013.



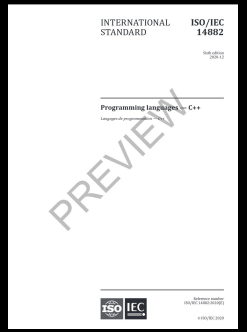
Gamma, E. Padrões de Projetos: Soluções Reutilizáveis. Bookman. 2009.



Deitel, H. M., Deitel, P. J. C++: como programar. 10a ed. Pearson Prentice Hall. 2017.



ISO/IEC 14882:2020 Programming languages - C++: www.iso.org/obp/ui/#iso:std:iso-iec:14882:ed-6:v1:en



Licença