

# Estilo de Código e STL

# Agenda

- Noções de Estilo de Código
- Uso do Switch
- STL

# Estilo de Código

- Escolha um e seja consistente
  - Linux
  - GNU
  - etc
- Principio Básico:
  - *Legibilidade*

# Legibilidade

- Nomes
  - inteligíveis
  - descritivos
  - Não peque pela falta nem pelo excesso

# Legibilidade

- Exemplos:

- RUIM:

- np\_branças

- numero\_de\_pecas\_branças\_no\_batuleiro

- BOM:

- num\_pecas\_branças

# Legibilidade

- Estilos para nomes
  - ALL\_CAPS
    - geralmente usado para constantes, macros
  - letras\_minúsculas
    - geralmente usada para variáveis locais, atributos e nomes de função
  - NotaçãoCamelo
    - Geralmente usada para nomear classes
  - mixedCase
    - Geralmente usada para nomear métodos

# Legibilidade

- Identação:

```
if ( condição ) {  
    ações  
} else if ( ) {  
    ações  
} else {  
    ações  
}
```

# Uso do switch

- Contando os pontos no buraco:

```
switch (carta)
  case ás:
    valor += 15;
    break;
  case valete:
  case dama:
  case rei:
  case 2:
    valor += 10;
    break;
  default:
    valor += 5;
}
```



Sentido da busca

# Uso do switch:

- Não esqueça do break:
  - ele evita da busca continuar.

# STL

- Standard Template Library
  - Biblioteca de “gabaritos” padrão do C++
- Oferece várias funcionalidades ao programador
  - Estruturas de dados já prontas ( e eficientes )
    - Listas, Vetores, Tabelas Hash, Pilhas, etc
  - Algoritmos ( busca, achar o maior elemento, for\_each, etc )

# STL

- Gabaritos

- Permite descrever estruturas, classes e funções de forma genérica

- Abstração de tipos

Inicialização



- Exemplo

- `Vector<Peça> peças_do_xadrez(32);`

Tipo do gabarito



Nome do gabarito



# STL

- Vectors

- Arrays que podem ser redimensionados

- Operações

- Adição de um elemento:

- `peças_do_xadrez.push_back(e);`

- Acesso a elementos

- `e = peças_do_xadrez[i];`

- `e = peças_do_xadrez.back();`

- `e = peças_do_xadrez.front();`

- Acesso verificado a elementos

- `e = peças_do_xadrez.at(i);`

- Tamanho

- `n = peças_do_xadrez.size();`

# STL

- Listas

- Listas encadeadas

- Operações

- Adição de um elemento:

- `peças_do_xadrez.push_back(e);`

- `peças_do_xadrez.push_front(e);`

- Remoção de um elemento:

- `e = peças_do_xadrez.pop_front();`

- `e = peças_do_xadrez.pop_back();`

- Tamanho

- `n = peças_do_xadrez.size();`

# STL

- Iteradores
  - Oferecem métodos unificados de percorrer listas, vectors e outras estruturas da STL

# STL

- Iteradores

- Exemplo:

```
typedef Vector<Peças> VecPeças;
typedef Vector<Peças>::const_iterator VecPeçasIterator;
void soma_valor( const VecPeças peças){
    int brancas = 0;
    int pretas = 0;
    for(VecPeçasIterator i = peças.begin(); i != peças.end
        ( ); i++){
        Peça& peça = *i: // Abreviar
        if ( peça.é_branca() ){
            brancas += peça.obtemValor();
        } else {
            pretas += peça.obtemValor();
        }
    }
}
```