

O ponteiro this

Paulo Ricardo Lisboa de Almeida

Editando makefile

- Edite o makefile
- Adicione o parâmetro de compilação `-Wshadow`
parametrosCompilacao=-Wall -Wshadow
- A opção `-Wshadow` avisa se criamos alguma variável ou função que pode ocultar outra variável ou função
 - Para detalhes, digite no terminal *man g++*, e procure por `-Wshadow`

Em Pessoa.hpp e Pessoa.cpp

- Modifique o nome do parâmetro do setCpf para *cpf*

Em Pessoa.hpp

```
void setCpf(unsigned long cpf);
```

Em Pessoa.cpp

```
void Pessoa::setCpf(unsigned long cpf){  
    if(validarCPF(cpf))  
        cpf = cpf;  
}
```

Em Pessoa.hpp e Pessoa.cpp

- Modifique o nome do parâmetro do setCpf para *cpf*
 - Algumas perguntas
 - Podemos fazer isso?

Em Pessoa.hpp

```
void setCpf(unsigned long cpf);
```

Em Pessoa.cpp

```
void Pessoa::setCpf(unsigned long cpf){  
    if(validarCPF(cpf))  
        cpf = cpf;  
}
```

Em Pessoa.hpp e Pessoa.cpp

- Modifique o nome do parâmetro do setCpf para *cpf*
 - Algumas perguntas
 - Podemos fazer isso?
 - Sim, podemos. Mas note que existe um dado membro chamado cpf.
 - Como cpf foi declarado dentro da função novamente, agora o dado membro será ignorado, e somente **a variável local cpf está sendo referenciada**.
 - O dado membro *cpf* está sendo sombreado (*shadowed*) pela *variável local cpf*

Em Pessoa.hpp

```
void setCpf(unsigned long cpf);
```

Em Pessoa.cpp

```
void Pessoa::setCpf(unsigned long cpf){  
    if(validarCPF(cpf))  
        cpf = cpf;  
}
```

Em Pessoa.hpp e Pessoa.cpp

- Modifique o nome do parâmetro do setCpf para *cpf*
 - Algumas perguntas
 - Podemos fazer isso?
 - Sim, podemos. Mas note que existe um dado membro chamado cpf.
 - Como cpf foi declarado dentro da função novamente, agora o dado membro será ignorado, e somente **a variável local cpf está sendo referenciada**.
 - Funciona?

Em Pessoa.hpp

```
void setCpf(unsigned long cpf);
```

Em Pessoa.cpp

```
void Pessoa::setCpf(unsigned long cpf){  
    if(validarCPF(cpf))  
        cpf = cpf;  
}
```

Em Pessoa.hpp e Pessoa.cpp

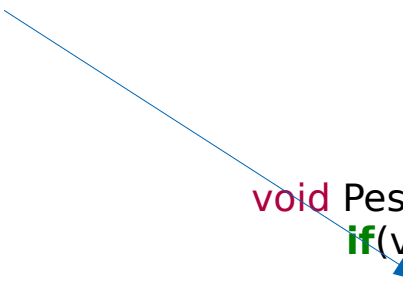
- Modifique o nome do parâmetro do setCpf para *cpf*
 - Algumas perguntas
 - Podemos fazer isso?
 - Sim, podemos. Mas note que existe um dado membro chamado cpf.
 - Como cpf foi declarado dentro da função novamente, agora o dado membro será ignorado, e somente **a variável local cpf está sendo referenciada**.
 - Funciona?
 - Do jeito que fizemos não
 - A variável local recebe o valor dela mesma?

Em Pessoa.hpp

```
void setCpf(unsigned long cpf);
```

Em Pessoa.cpp

```
void Pessoa::setCpf(unsigned long cpf){  
    if(validarCPF(cpf))  
        cpf = cpf;  
}
```



Teste você mesmo

- Limpe os arquivos objeto e compile novamente
make clean
make
- Veja que o -Wshadow nos avisou da besteira que estamos fazendo

... warning: **declaration of 'cpf' shadows a member of 'Pessoa'** [-Wshadow]

```
void Pessoa::setCpf(unsigned long cpf){  
    ^
```

...

Pessoa.hpp:28:17: note: **shadowed declaration is here**

```
    unsigned long cpf;  
    ^~~
```

...

Em Pessoa.hpp e Pessoa.cpp

- Desejamos uma forma de dizer ao compilador a “qual cpf” estamos nos referenciando

```
void Pessoa::setCpf(unsigned long cpf){  
    if(validarCPF(cpf))  
        cpf = cpf;  
}
```

Aqui, desejamos o **dado membro** cpf

Aqui, desejamos o **parâmetro** cpf

O Ponteiro *this*

- O ponteiro especial *this* aponta para o **objeto** atual na memória
- Em outras palavras, **contém o endereço do objeto atual**
- O ponteiro é implicitamente “declarado” para todos os objetos
 - Todo objeto têm seu próprio *this* (você não precisa declarar)

Teste você mesmo

- Crie uma função membro em Pessoa que imprime o endereço de memória do objeto Pessoa instanciado atualmente

Em Pessoa.hpp

```
void imprimeEnderecoMemoria();
```

No main.cpp

```
int main(){  
    Pessoa p1{"Joao", 30};  
    Pessoa p2{"Maria", 20};  
    p1.imprimeEnderecoMemoria();  
    p2.imprimeEnderecoMemoria();  
    return 0;  
}
```

Em Pessoa.cpp

```
void Pessoa::imprimeEnderecoMemoria(){  
    std::cout << this << std::endl;  
}
```



O cout vai identificar que this é um ponteiro,
e vai imprimir o endereço em hexa na tela

Usando o this para evitar sombreamentos

- Como utilizar o this para evitar os problemas que criamos na função membro setCpf?

```
void Pessoa::setCpf(unsigned long cpf){  
    if(validarCPF(cpf))  
        cpf = cpf;  
}
```

Usando o *this* para evitar sombreamentos

- Como utilizar o *this* para evitar os problemas que criamos na função membro `setCpf`?
 - Basta adicionar *this* na frente do `cpf` que desejamos que aponte para o dado membro
 - Lembre que para acessar um membro via ponteiro, da mesma forma que em C, utilizamos o operador `->`

Usando o this para evitar sombreamentos

Pessoa.hpp

Pessoa.cpp

```
class Pessoa{  
public:  
    //...  
    void setCpf(unsigned long cpf);  
  
    void imprimeEnderecoMemoria();  
private:  
    bool validarCPF(unsigned long cpfTeste);  
  
    std::string nome;  
    unsigned long cpf;  
    unsigned char idade;  
};
```

```
void Pessoa::setCpf(unsigned long cpf){  
    if(validarCPF(cpf))  
        this->cpf = cpf;  
}
```

Se refere ao dado membro cpf

#endif

Dica

- Remova o -Wshadow dos parâmetros de compilação
 - Caso contrário o compilador vai continuar reclamando devido ao nome do parâmetro ser o mesmo do dado membro
 - Já resolvemos o problema através do this->

Boas práticas de programação

- Nos sets e construtores sempre utilize como nomes de parâmetros o **mesmo nome dos dados membro**
 - Evita a proliferação de identificadores
 - Use o ponteiro **this** para identificar quando você está se referindo aos dados membro
- **Sempre use o this nas funções membro quando se referenciar a um dado membro**, mesmo que não haja uma variável fazendo sombreamento

Dica

- O ponteiro `this` existe em diversas linguagens orientadas a objetos, e seu uso é amplamente encorajado
 - Exemplos em Java e C#
 - Note que em Java e C# não existem ponteiros, mas o uso do *this* é similar
 - https://www.w3schools.com/java/ref_keyword_this.asp
 - <https://docs.microsoft.com/pt-br/dotnet/csharp/language-reference/keywords/this>

Exercício

- Modifique todos os sets e construtores das classes para que os parâmetros possuam o mesmo nome dos dados membro
 - Indique o acesso de dados membro através do ponteiro *this* em todas as funções membro
 - Nos construtores, caso você utilize a lista de inicializador de membro, **não utilize o this**
 - O compilador já sabe que o conteúdo que está entre chaves se refere ao parâmetro, e o que está fora das chaves é o dado membro
 - Exemplo

```
Pessoa::Pessoa(std::string nome)
    :nome{nome} {
}
```

Referências

- DEITEL, P.; DEITEL, H. **C++ how to Program**. [S.l.]: Pearson, 2017. ISBN 9780134448237
- STROUSTRUP, B. **The C++ Programming Language**. Pearson Education, 2013. ISBN 9780133522853.