

II. Encapsulamento

Arquivos .h e .cpp

Propriedades

Getters / Setters

Classe Data

Aa

Separando uma classe

.H

- Atributos
- Protótipo dos métodos

.CPP

- Implementação dos métodos

MAIN

- Teste da classe

Métodos Inline

- Incluir seu código diretamente na declaração da classe
- Implicações:
 - substitui as mensagens por uma cópia do código
 - código torna-se mais rápido (bom)
 - código torna-se maior (ruim)
 - código torna-se acessível (ruim)
- Também funciona utilizando-se a palavra reservada “inline” antes do nome do método

Exemplo

```
class Contador
{
    int num;
    public:
    Contador() {num=0;}
    void incrementa() {num=num+1;} //inline
    int getNum() {return num;}
};
```

Exemplo

```
class Contador {  
    int num;  
    public:  
        Contador() {num=0;}  
        int getNum() {return num;}  
};
```

```
inline void Contador::incrementa() { //inline  
    num=num+1;  
}
```

SELETORES E MODIFICADORES

Getters and Setters

- Métodos que fazem o “encapsulamento” de atributos “privados”.
- **Getter**: retorna o atributo
- **Setter**: modifica o atributo
- É a única forma de referenciar um atributo privado fora da classe.
- Permitem controle adicional, além da atribuição/acesso propriamente dito.

Getters and Setters

- Getter padrão
 - atributo tipoX abc;
 - tipoX getAbc() { return abc;}
- Setter padrão
 - atributo tipoX abc;
 - void setAbc(tipoX aux) { abc = aux;}

Exemplo

```
class Func {  
    private:  
        int salario;  
  
    public:  
        // Setter  
        void setSalario(int s) {  
            salario = s;  
        }  
        // Getter  
        int getSalario() {  
            return salario;  
        }  
};
```

```
int main() {  
    Func f1;  
    f1.setSalario(50000);  
    cout << f1.getSalario();  
    return 0;  
}
```

Getters e Setters

- CUIDADO!!
- Nem todo atributo deve conter seletores e modificadores
 - int código;
 - Deve ter GET
 - Mas quem pode alterar um código?

Nem todos os atributos precisam de
getters e setters

Getters e Setters

- Por que devemos usar métodos?
 - CONTROLE

```
void setMes(int m) {  
    if(m>0 and m<13)  
        this->mes = m;  
    else  
        printf("mês invalido");  
}
```

Exemplo

```
class Func {  
    private:  
        int codigo;  
        int salario;  
  
    public:  
        // Setter  
        void setSalario(int s, string login) {  
            if( login == "admin")  
                salario = s;  
        }  
        // Getter  
        int getCodigo() { return codigo; }  
        int getSalario() { return salario; }  
};
```

EXERCÍCIOS

Classe CAIXA

- Separar a classe em arquivos C e H
- Adicionar métodos Seletores e Modificadores que sejam Inline.
- Crie um campo Código (sem Setter)
- Treine a questão do encapsulamento fazendo outras checagens legais nos métodos setters.
 - altura, largura e profundidade não podem ser negativos
 - aceite peso em grama ou em quilos e converta para quilos

Vai estar no ED!!