

# Programação Orientada a Objetos

Unidade 6 – Classes Abstratas

# Relembrando...



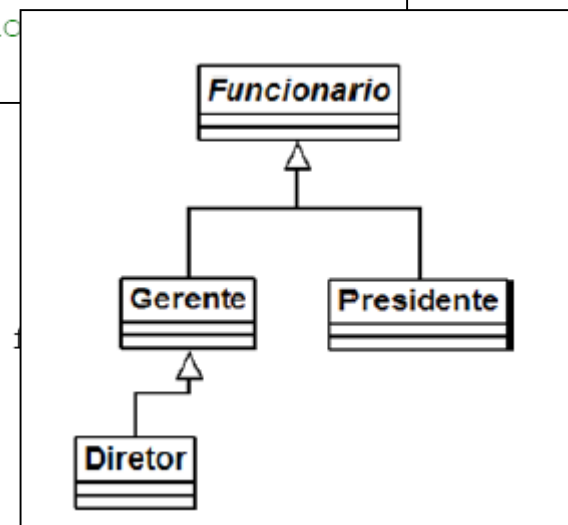
```
public class Funcionario
{
    protected String _nome;
    protected String _cpf;
    protected double _salario;

    public double BuscaBonificacao()
    {
        return this._salario * 1.2;
    }
    // outros métodos
}
```

```
public class ControleDeBonificacoes
{
    private double _totalDeBonificacoes = 0;

    public void Registra(Funcionario f)
    {
        Console.WriteLine
            ("Adicionando bonificacao do funcionario: " + f._nome);
        this._totalDeBonificacoes += f.BuscaBonificacao();
    }

    public double TotalDeBonificacoes
    {
        get{ return this._totalDeBonificacoes; }
    }
}
```



# Perguntas



Faz sentido ter uma referência de funcionário?

Faz sentido ter um objeto funcionário?

# Faz sentido?



```
ControleDeBonificacoes cdb = new ControleDeBonificacoes();  
Funcionario f = new Funcionario();  
cdb.Adiciona(f); // faz sentido?
```

# Classe Abstrata



O que, exatamente, vem a ser a nossa classe Funcionario?  
Nossa empresa tem apenas Diretores, Gerentes, Secretárias,  
etc.

Ela é uma classe que apenas idealiza um tipo, define apenas um  
rascunho.

# Classe Abstrata



Usamos a palavra chave **abstract** para impedir que ela possa ser instanciada. Esse é o efeito direto de se usar o modificador **abstract** na declaração de uma classe:

```
public abstract class Funcionario
{
    protected double _salario;

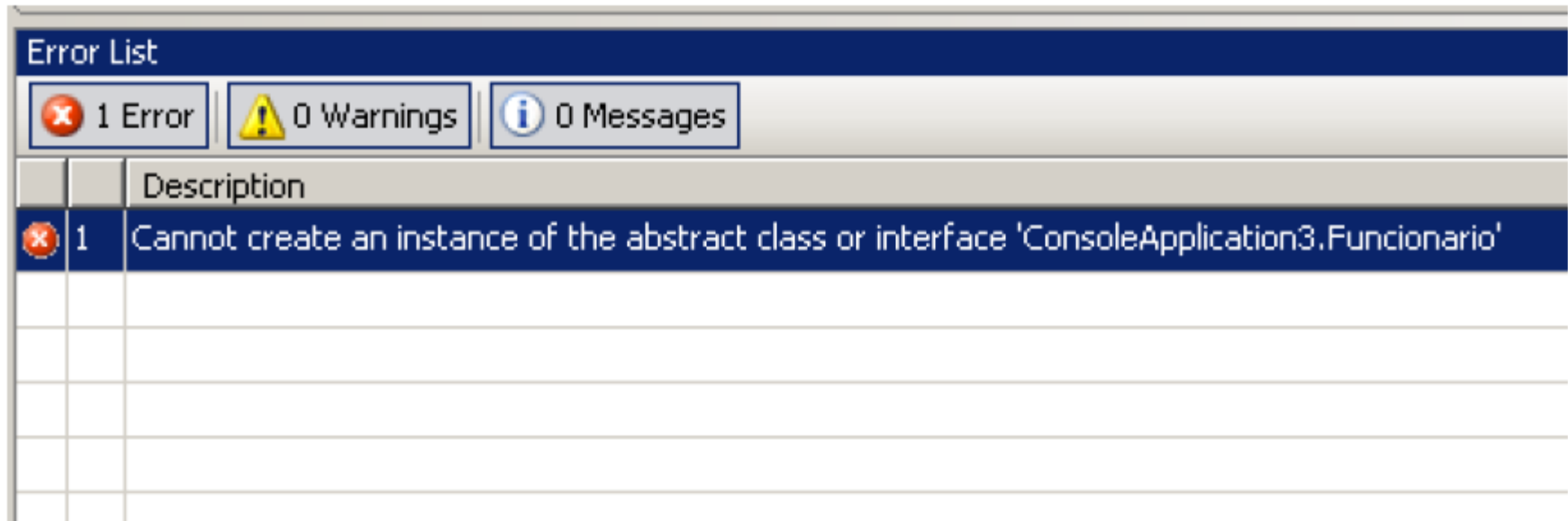
    public double BuscaBonificacao()
    {
        return this._salario * 1.2;
    }
    // outros atributos e metodos comuns a todos Funcionarios
}
```

# Classe Abstrata



E, no meio de um código:

```
Funcionario f = new Funcionario();
```



# Métodos Abstratos



Levando em consideração que cada funcionário em nosso sistema tem uma regra totalmente diferente para ser bonificado, faz algum sentido ter esse método na classe Funcionario?



# Métodos Abstratos



Poderíamos, então, jogar  
fora esse método da classe  
Funcionario?

# Métodos Abstratos



```
public abstract class Funcionario
{
    protected double _salario;

    public abstract double BuscaBonificacao();

    // outros atributos e metodos comuns a todos Funcionarios}
}
```

# Programação Orientada a Objetos

Unidade 6 – Classes Abstratas