



# IFRN

Tecnologia em Análise e Desenvolvimento de  
Sistemas

---

## Classes e Objetos em C#

Prof. Gilbert Azevedo



# Conteúdo

---

- Entidade
- Classificação
- Classe e Objeto
- Definição de Classe em C#
- Encapsulamento
- Acessibilidade
- Operador New
- Construtores



# Entidades

---

- As entidades representam conjuntos de objetos da realidade modelados em um sistema computacional
  - Ex: Clientes, Produtos, Vendas, etc
- Na modelagem de um sistema, identificam-se as identidades relevantes, as informações que elas armazenam e as funções que elas executam
  - Em um sistema bancário, a entidade conta-corrente armazena dados do cliente, o saldo e o número da conta e realiza as operações de débito e crédito
- Na POO, estruturas denominada classes são utilizadas para modelar as entidades manipuladas no aplicativo
  - O armazenamento de informações é realizado nos campos
  - A execução de suas funções é feita através de métodos

# Classificação

---

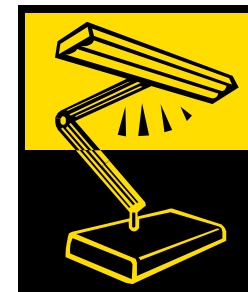
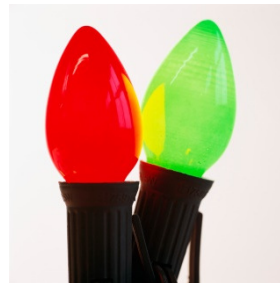
- No projeto uma classe, organiza-se sistematicamente as informações e funcionalidades de uma entidade
- A organização é um ato de classificação realizado por todas as pessoas, desde crianças, não apenas por programadores
- As pessoas usam a classificação para nomear todas as coisas que possuem comportamentos e atributos comuns
- Carros, livros, relógios, etc, são exemplos de classificação



# Exemplo de Classificação

---

- O termo “Lâmpada” é utilizado para identificar objetos utilizados na iluminação de um ambiente
  - Atributos comuns: potência, voltagem, cor, estado
  - Comportamentos comuns: acender, apagar
- Em um modelo computacional, a classe Lâmpada define os atributos e comportamentos comuns a todos esses objetos
- Cada lâmpada é um objeto ou instância desta classe





# Classe e Objeto

---

- Classe

- É a unidade de software da POO que descreve uma entidade
- Lista as características da entidade (Atributos)
- Lista o comportamento da entidade (Métodos)
- Define um tipo (de variável)

- Objeto

- É uma variável da classe
- Armazena valores em cada atributo (Estado)
- Executa as funções definidas nos métodos (Comportamento)
- Ocupa um local de memória que o identifica unicamente (Identidade)
- É acessado por uma referência



# Definindo uma classe em C#

---

- A palavra-chave *class* é utilizada para definir uma nova classe
- A classe Lâmpada, por exemplo, representa uma entidade lâmpada
- Cada lâmpada (instância) armazena o seu estado (ligado ou desligado) e realiza as operações Ligar, Desligar e EstaLigada

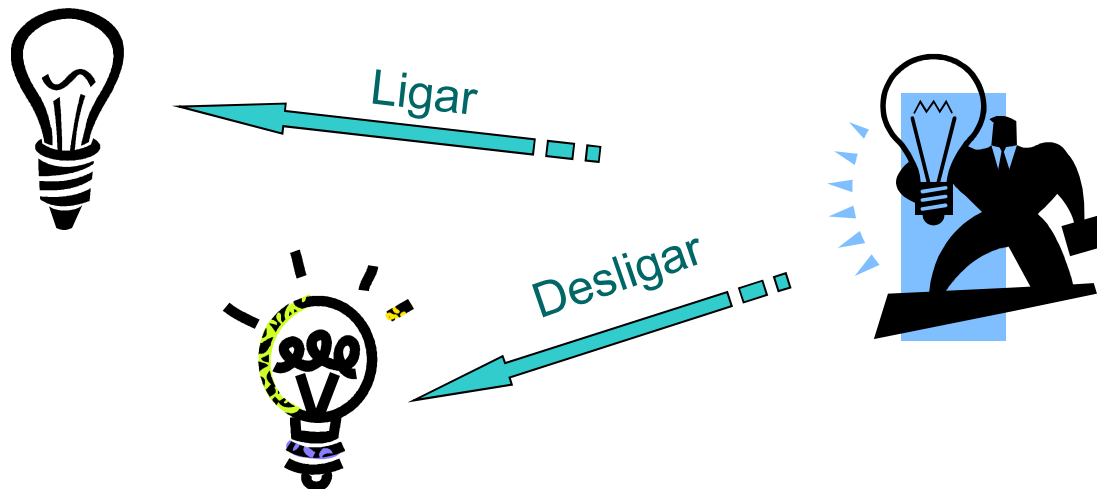
class Lampada

```
{  
    bool ligada;  
    void Ligar() { ligada = true; }  
    void Desligar() { ligada = false; }  
    bool EstaLigada() { return ligada; }  
}
```

Lampada
- ligada : bool
+ Ligar() : void + Desligar() : void + EstaLigada() : bool

# Encapsulamento

- O encapsulamento é um princípio importante da POO
  - A ideia é “esconder” do usuário da classe detalhes da sua implementação
  - O usuário instancia a classe, obtendo um objeto, e utiliza seus métodos sem se preocupar como eles foram implementados
  - Na prática, os métodos da classe são visíveis e os campos são encapsulados (envolvidos em uma cápsula) e inacessíveis



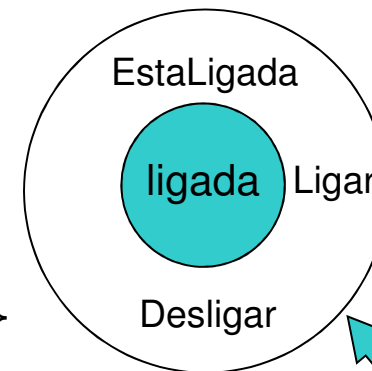


# Acessibilidade

- Para realizar o encapsulamento são utilizados modificadores de acessibilidade
- O modificador *private* deve ser utilizado nos campos da classe, tornando-os inacessíveis fora desta
- O modificador *public* deve ser utilizado nos métodos visíveis ao mundo externo

class Lampada

```
{  
    private bool ligada;  
    public void Ligar() { ligada = true; }  
    public void Desligar() { ligada = false; }  
    public bool EstaLigada() { return ligada; }  
}
```





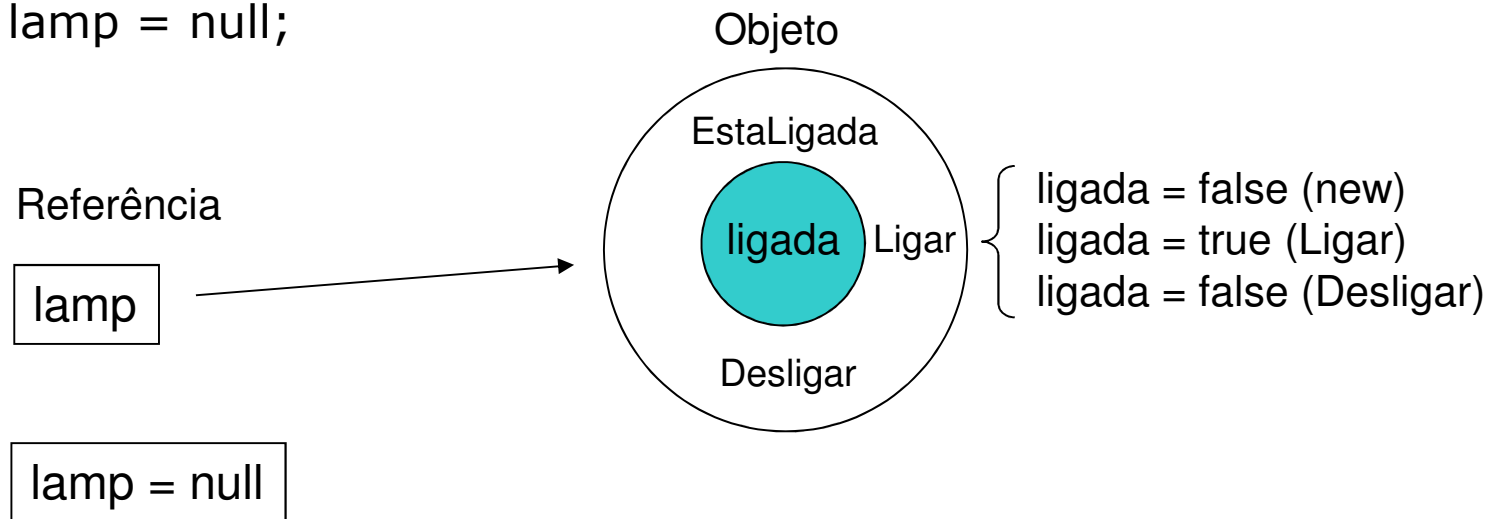
# Operador *New*

---

- O operador *new* é utilizado para instanciar uma classe, criando um objeto (variável ou instância da classe).
- Os objetos são controlados por uma referência.
- Classe Referência = *new* Construtor();
  - A classe define o tipo de objeto que é criado.
  - A referência é a variável que controla o objeto.
  - O construtor é um método especial (com o mesmo nome da classe) que aloca o objeto na memória.
- A palavra reservada *null* é utilizada para indicar que uma referência não controla nenhum objeto.
  - Referência = *null*;

# Valores Padrões de Atributos

- Os atributos de um objeto são automaticamente iniciados com 0, *false* ou *null*, dependendo do seu tipo.
- Lampada lamp = new Lampada();
- lamp.Ligar ();
- lamp.Desligar();
- lamp = null;





# Sobrecarga de Construtores

---

- Quando uma classe é declarada, o C# fornece um construtor padrão (sem parâmetros) para a classe
- Outros construtores podem ser definidos para realizar uma inicialização personalizada
- Uma classe pode conter vários construtores desde que suas listas de parâmetros os identifique unicamente
- Quando algum construtor é definido, o construtor padrão não é mais fornecido pelo C#. Se necessário, ele deve ser escrito explicitamente
- A ordem de escrita dos construtores é irrelevante
- A palavra reservada *this* pode ser utilizada para realizar a chamada entre construtores



# Exemplo de Construtores

---

- class Lampada
- {
- private bool ligada;
- public Lampada() : this(true)
- {
- }
- public Lampada(bool aLigada)
- {
- ligada = aLigada;
- }
- public void Ligar() { ligada = true; }
- public void Desligar() { ligada = false; }
- public bool EstaLigada() { return ligada; }
- }



# Exercícios

---

- Desenvolver classes para representar:
- 1. Um retângulo que armazene suas dimensões (base e altura) e calcule a sua área.
- 2. Uma conta corrente bancária com informações de nome do cliente, número da conta e saldo. A conta deve realizar as operações de débito, crédito e verificação de saldo.



# Referencias Bibliográficas

---

- Introduction to C# Programming with Microsoft .Net
  - Microsoft Official Course 2609A
- Microsoft Visual C# 2005 - Passo a passo
  - John Sharp, Bookman, 2007
- Microsoft Asp.Net – Passo a passo
  - George Sheperd, Bookman, 2007
- Microsoft VS 2005 Express Edition Documentation
- POO – Programação Orientada a Objetos
  - Prof. Plácido Neto - IFRN