

# Classes Abstratas x Interfaces

Prof. Marcelo Cohen

## 1. Revisão: Classes Abstratas

- Permitem que se definam métodos sem implementação que devem ser redefinidos em classes derivadas.
- Não podem ser instanciadas.
- O que define se uma classe é abstrata ou não, é a ocorrência de pelo menos um método abstrato.

```
abstract class Figura
{
    protected int x,y;

    public Figura(int x,int y)
    {
        this.x = x;
        this.y = y;
    }

    abstract public void desenha();
}
```

## 2. Interfaces

- Interface
  - Palavra-chave **interface**
  - Define um conjunto de métodos **public abstract**
  - Pode definir atributos **public final static**
- Usando interfaces
  - Uma classe específica que vai usar uma interface com a palavra-chave **implements**
    - Pode-se implementar várias interfaces, escrevendo uma lista separada por vírgulas
  - Classe deve definir todos os métodos **abstract** na interface
    - Devem usar o mesmo número de argumentos e tipo de retorno
  - Usar uma interface é como “assinar um contrato”
    - “Eu vou implementar todos os métodos especificados na interface”

## Exemplo de Interface

- Interface “Anda”

```
public interface Anda
{
    void avancar(int deslx,int desly);
    void recuar (int deslx,int desly);
}
```

- Define um objeto que “anda” (no solo, em qualquer direção)
  - Se uma classe implementar essa interface, deverá conter o código para os métodos `avancar(...)` e `recuar(...)`

## Exemplo de Interface (2)

- Interface “Voa”

```
public interface Voa
{
    void subir (int desl_altura);
    void descer(int desl_altura);
}
```

- Define um objeto que “voa” (para cima e para baixo)
  - Se uma classe implementar essa interface, deverá conter o código para os métodos subir(...) e descer(...)

## Exemplo de Interface (3)

- Classe Tartaruga

- Tartarugas, em geral, não voam. Portanto, iremos apenas implementar a primeira interface:

```
public class Tartaruga implements Anda
{
    private int posx, posy; // posição da tartaruga

    public void avancar(int deslx, int desly)
    {
        posx += deslx;
        posy += desly;
    }

    public void recuar(int deslx, int desly)
    {
        ...
    }
}
```

## Exemplo de Interface (4)

- Classe BemTeVi
  - Bem-te-vis normalmente voam. Mas também caminham no solo. Então, iremos implementar ambas interfaces:

```
public class BemTeVi implements Anda, Voa
{
    private int posX, posY; // posição do bem-te-vi (no solo)
    private int altura;     // altura do bem-te-vi

    public void avancar(int deslx, int desly) { ... }
    public void recuar(int deslx, int desly) { ... }

    public void subir(int desl_alt) { altura += desl_alt; }

    public void descer(int desl_alt) { altura -= desl_alt; }
}
```