



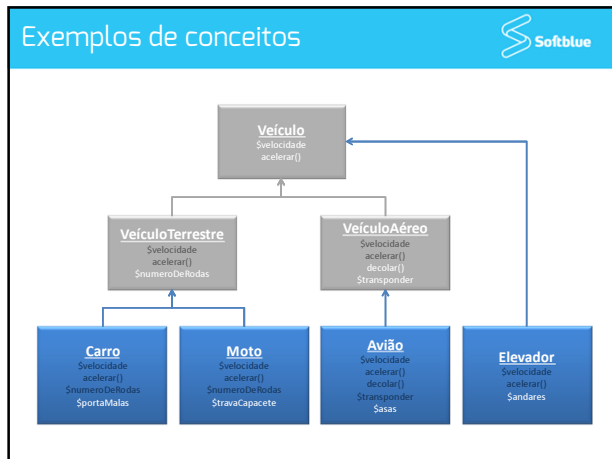


Abstração 

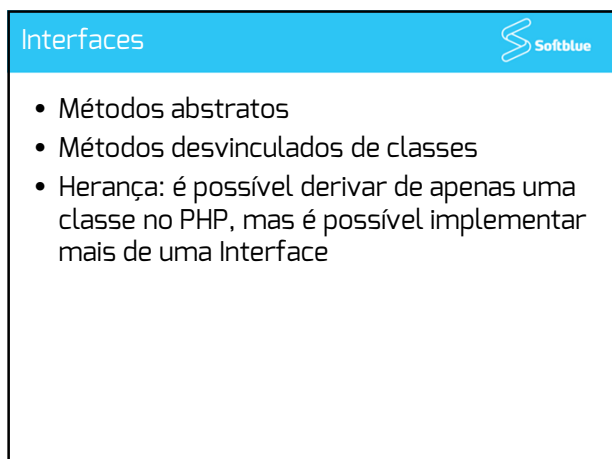
- Virtualização de objetos-conceitos
 - Exemplos: veículo, cor, móvel, ser vivo
- Abstração de um conceito que as subclasses irão implementar
- Padronização: evita que cada subclasse crie sua própria assinatura do método
- Facilita a utilização em laços de repetição e navegação de listas de objetos
- É uma classe não instanciável

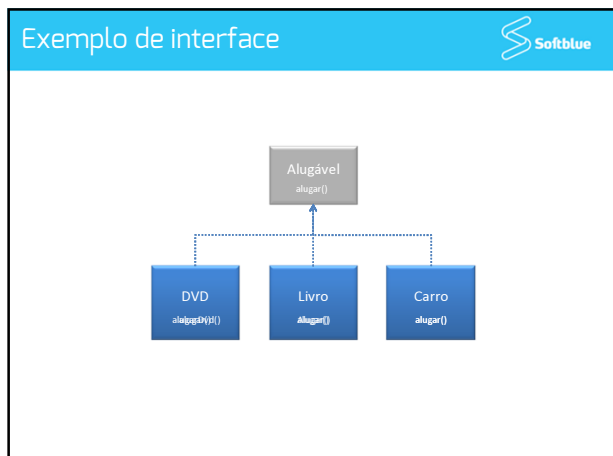
Abstração 

- Somente classes abstratas podem conter métodos abstratos
- Classes abstratas podem conter atributos e métodos não abstratos
- A primeira classe não-abstrata deve implementar as assinaturas dos métodos abstratos da classe de qual deriva
- Classes abstratas podem herdar de outras classes abstratas sem implementar os métodos









Interfaces na prática

- Operador **interface**

```

interface Alugavel {
    public function alugar();
    public function retornarLocacao();
}
  
```

- Operador **implements**

```

class Livro implements Alugavel {
    public function alugar() {
        // implementação
    }
    public function retornarLocacao() {
        // implementação
    }
}
  
```

```

class Carro extends VeiculoTerrestre implements Alugavel {
    public function acelerar() {
        // implementação
    }
    public function alugar() {
        // implementação
    }
    public function retornarLocacao() {
        // implementação
    }
}
  
```

Interfaces

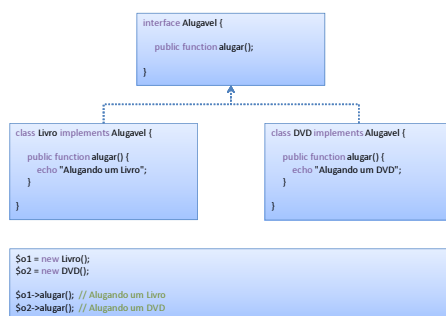
- Geralmente nomes que indicam uma qualidade
- Classes abstratas podem implementar interfaces, e podem ou não implementar os métodos
- A primeira classe não abstrata de uma hierarquia deve implementar ou herdar implementados todos os métodos abstratos
- Não é possível incluir atributos em interfaces
- Uma classe pode implementar uma ou mais interfaces, mas herdar de uma classe no máximo
 - `implements Alugavel, Dirigivel, Deletavel`

Polimorfismo

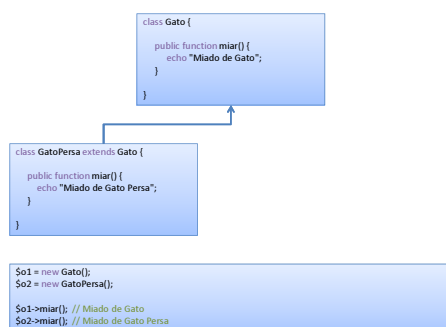


- Capacidade de um mesmo método executar diferentes comportamentos, dependendo do objeto sobre o qual está sendo invocado
- Não possui operador pois é um conceito
- Situações mais comuns que geram polimorfismo:
 - Sobrescrita de métodos
 - Implementação de classes abstratas
 - Implementação de interfaces


Exemplo de polimorfismo



Exemplo de polimorfismo

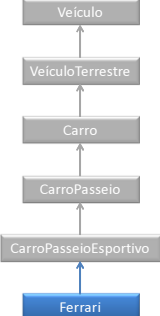


Considerações finais sobre OO





- Benefícios
- É uma tendência
- Uso no mercado cada vez maior
- Abstrair a realidade até que ponto para meu projeto?
- Coesão

Abstração da realidade deve ser feita levando em conta as características de cada projeto, abstraindo somente os dados necessários.



```
graph BT; Ferrari --> CarroPasseioEsportivo; CarroPasseioEsportivo --> CarroPasseio; CarroPasseio --> Carro; Carro --> VeiculoTerrestre; VeiculoTerrestre --> Veiculo;
```

Aulas práticas e manuais on-line



Assista agora as aulas práticas.

[Clique aqui](#) para visualizar as aulas práticas disponíveis
