

# Java I

## Programando para interfaces

Prof. Sanderson Macedo  
[twitter.com/sandecom](https://twitter.com/sandecom)

# O que vamos ver hoje?

- Programando para interfaces;
- Contratos de implementação;
- Polimorfismo.



# Programando para interfaces

- Interface é a forma pelo qual uma classe se comunica com o mundo exterior;
- Todas as classes tem interfaces de comunicação;
- As principais interfaces de uma classe são os métodos GETTERS e SETTERS;
- A interface de uma classe é expressa pelos **membros públicos** de uma classe.

# Programando para Interface

- O princípio da interface é um alicerce da programação modular;
- Na programação orientada a objeto, a interface de um objeto consiste de um conjunto de métodos que um objeto deve suportar;
- Interface de um tipo são todos os seus membros públicos.

# Programando para Interface

- A Interface de um tipo são todas as suas operações públicas.



# Interface em Java

```
public interface MinhaInterface {  
    public void fazAlgo();  
}
```

# Outra interface

```
public interface InterfaceBacana {  
    public void fazOutraCoisa();  
}
```

# Implementando a interfaces

```
public class MinhaClasse implements MinhaInterface, InterfaceBacana {  
  
    @Override  
    public void fazAlgo() {  
  
    }  
  
    @Override  
    public void fazOutraCoisa() {  
  
    }  
  
}
```



# Interface

- Todos os métodos são abstratos;
- Não existe método construtor;
- Não existe atributos;

# O que não é interface?

- Qualquer membro da classe que não seja público;
- Variáveis de instância não fazem parte da interface de um objeto pois devem ser acessadas somente pelos "métodos de acesso".  
(encapsulamento)

O problema do Ben 10

# BEN 10



# Alguns personagens que Ben10 pode ser



Chama



XLR8



Massa  
Cinzenta

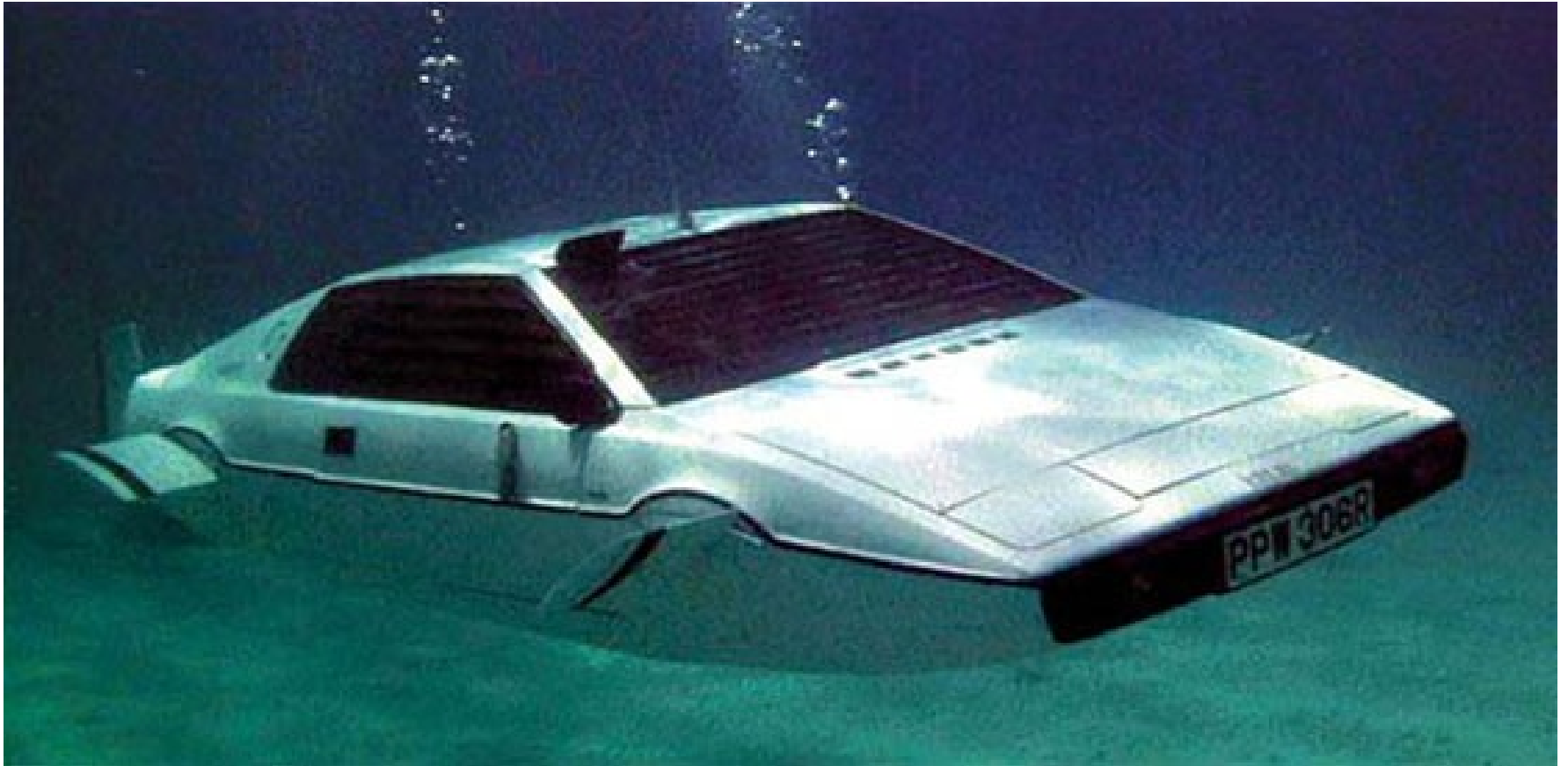


Quatro  
Braços

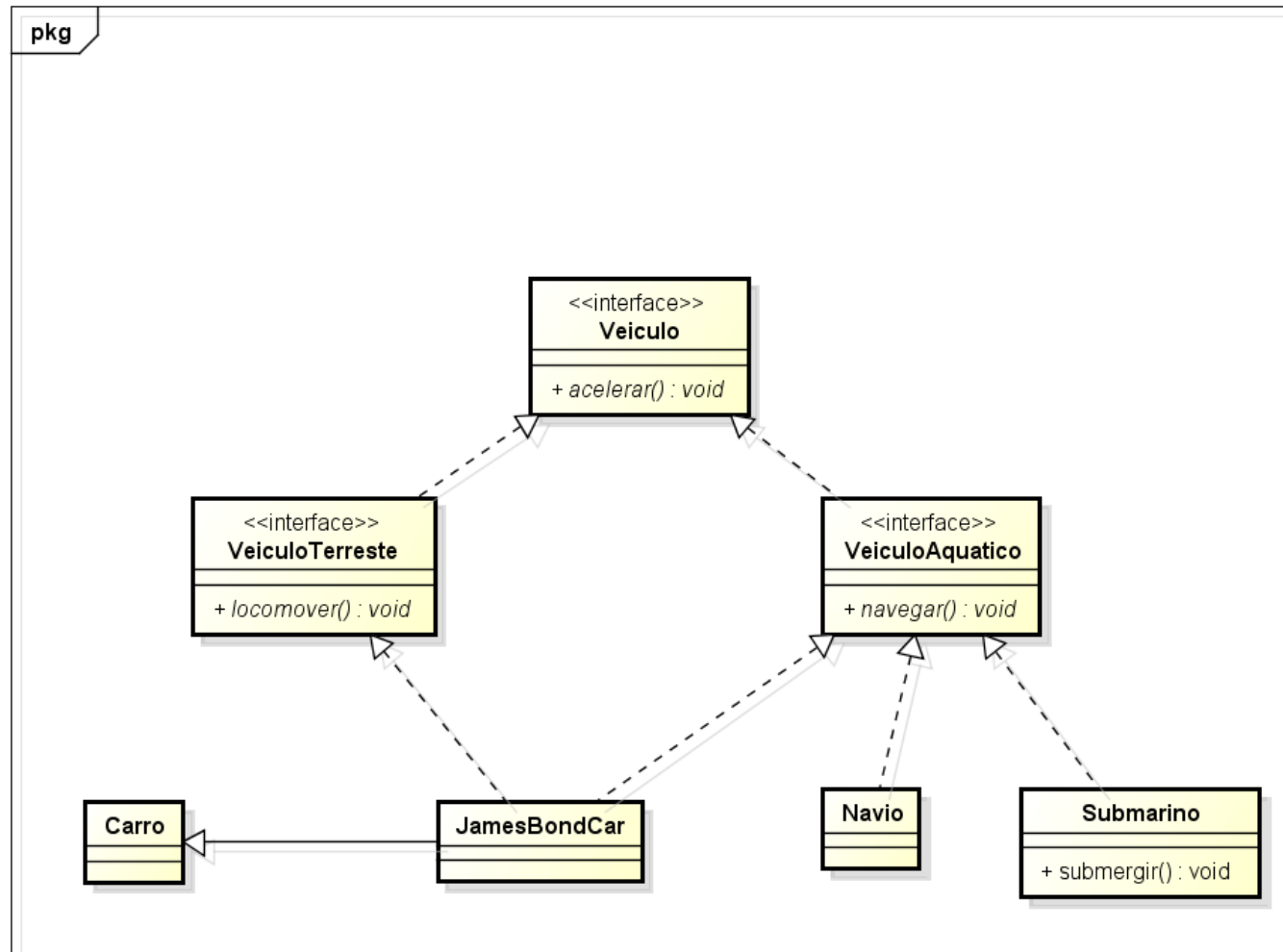


Ultra-T

# Problema do carro do James Bond



# Problema do Carro do James Bond



# Liberdade para trabalhar

- Interface é uma característica que permite a construção de formas de comunicação;
- Possibilita a criação de
- grupos de trabalho para
- desenvolvedores de um
- mesmo software;





# Contratos de implementação

- As interfaces não dizem como são implementados os métodos;
- As interfaces são contratos de que todos os seus métodos serão implementados por alguma classe;



# Contratos de implementação

- Por exemplo a classe BotECA não tem nenhuma implementação. Entretanto todas as classes que a estende, implementarão seus métodos de interface;
- **Isso é uma garantia dada, é um contrato!**
- Contratos de operação é obtido por meio do polimorfismo.



# Polimorfismo

O termo **polimorfismo** é originário do grego e significa "muitas formas"

- (poli = muitas, morphos = formas).



# Polimorfismo

- Propriedade pela qual duas ou mais classes derivadas de uma mesma superclasse ou que implementam uma interface;
- Podem invocar métodos que tem a mesma assinatura;
- Mas com comportamentos distintos.



# Polimorfismo

- Na programação orientada a objetos, o polimorfismo permite que referências de tipos mais abstratos representem o comportamento das classes concretas que referenciam;
- Assim, é possível tratar vários tipos de maneira homogênea (através da interface do tipo mais abstrato);

# Polimorfismo

- Por causa do polimorfismo é possível criar objetos da seguinte forma:

```
Boteca boteca = new Boteca();  
Boteca boteca1 = new Munheca();  
Boteca boteca2 = new Meleca();  
Boteca boteca3 = new Soneca();  
Boteca boteca4 = new Sapeca();
```

# Polimorfismo

- O uso de um objeto polimórfico obedecerá sempre a interface de criação (classe pai ou interface), mesmo que a classe implementadora tenha outros métodos que não foram declarados na interface;

# Polimorfismo

- No polimorfismo os métodos na classe filha que excede a interface da classe pai não estarão disponíveis.