

Interfaces

- ✓ Recurso que define ações que devem ser, obrigatoriamente, implementadas (cada classe pode executar de forma diferente)
- ✓ Um tipo de contrato firmado pela classe. A interface especifica o que deve ser implementado, mas, não como isso deve ser feito.
- ✓ Em UML se diz que uma classe realiza (realizes) uma interface.
- ✓ Contém apenas valores constantes (final).

Interfaces

- ✓ Não possui implementação, apenas a definição de métodos sem o corpo.
- ✓ Uma classe pode implementar diversas interfaces diferentes. Uma interface pode estender outra interface.
- ✓ Todos os métodos de uma interface são implicitamente públicos e abstratos.

Interfaces

- ✓ Não há como fazer uma instância de uma Interface e nem como definir um método construtor para a interface.
- ✓ As interfaces podem ser usadas para simular a herança múltipla.
- ✓ Ao incluir um novo método em uma interface será necessário ajustar todas as implementações dessa interface.

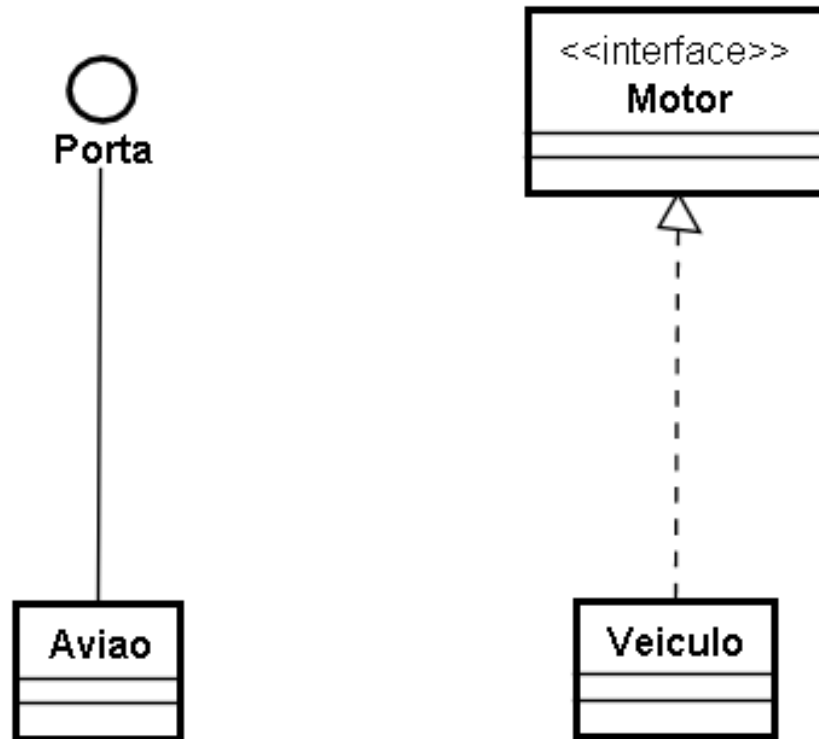
Uso de Interfaces - benefícios

- ✓ Padronizar a forma de acesso a um serviço
- ✓ Substituir uma classe por outra sem que o cliente note
- ✓ Criação de polimorfismo – objetos com métodos iguais, mas comportamentos diferentes
- ✓ Possibilidade de implementar composição em Java
- ✓ Reduzir o acoplamento

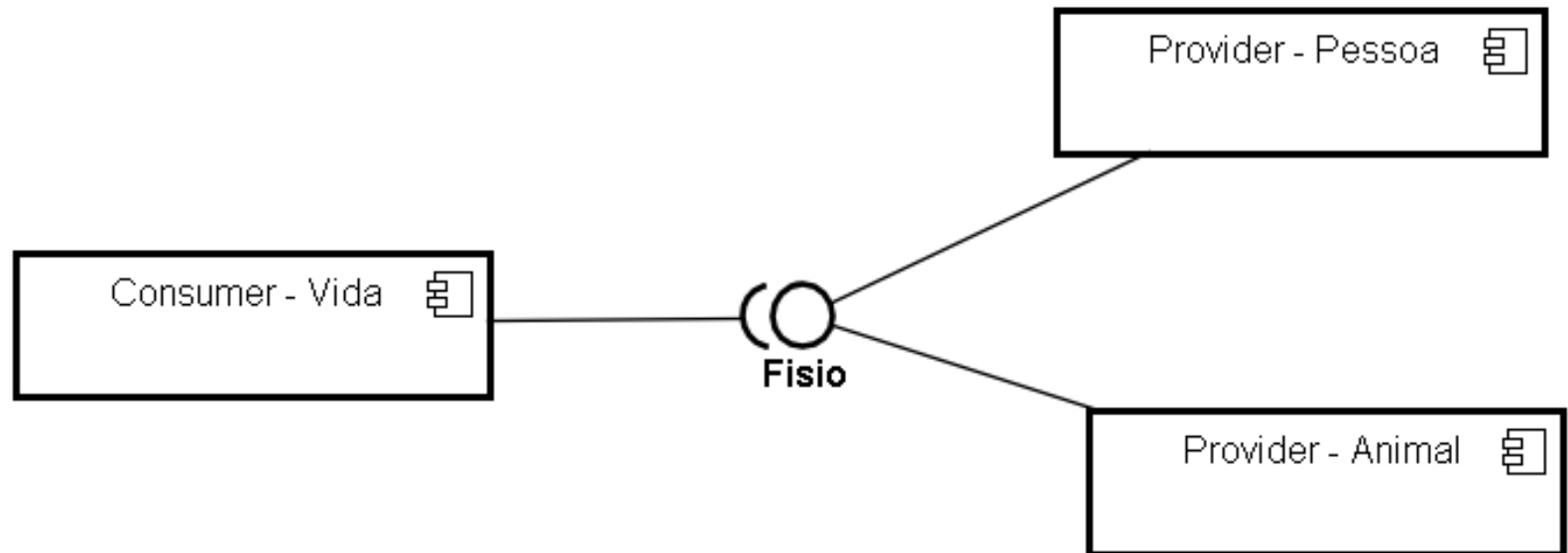
Programação Orientada a Objetos

Prof. Sérgio Furgeri – sergio.furgeri@fatec.sp.gov.br

Representação em UML



Em UML: Provided and Required Interface



Classes Abstratas

- ✓ Se todos os métodos da Classe abstrata forem sem corpo, ela funciona como uma Interface. Mas, ao contrário de uma interface, uma classe abstrata permite implementação de métodos.
- ✓ Assim como uma interface, é um tipo especial de classe que não há como criar instâncias dela.
- ✓ Classes abstratas podem conter declarações abstratas, concretas ou ambas.

Classes Abstratas

- ✓ Pode ser usada apenas para ser herdada, funciona como uma super classe. Isso força a hierarquia para todas as sub-classes.
- ✓ Para herdar uma classe abstrata é usada a palavra reservada **extends**, uma interface é implementada por meio da palavra reservada **implements**.
- ✓ Assim como uma interface, também pode atuar como um tipo de contrato que faz com que as sub-classes contemplem as mesmas hierarquias e/ou padrões.

Classes Abstratas

- ✓ Ao incluir um novo método em uma classe abstrata existe a opção de fornecer uma implementação padrão para ela, ou definir o método como abstrato e forçar a implementação nas subclasses.
- ✓ Os membros de uma classe abstrata podem conter qualquer visibilidade (pública, privada etc.), ao contrário de uma interface em que todos os membros são públicos.

Classes Abstratas

- ✓ Uma classe pode estender uma classe abstrata e implementar múltiplas interfaces.
- ✓ Exemplo: caso diversas classes usem a mesma implementação para os métodos save, update e remove, mas os métodos find são específicos para cada classe, é indicado criar uma classe abstrata ao invés da interface. Assim, implementaria os métodos save, update e remove, uma única vez para todas as classes, mas deixaria o método find como abstrato, para que cada classe fosse obrigada a criar sua própria implementação para esse método.

Exercícios

1. Se uma classe realizar a interface `java.lang.CharSequence`, quais métodos ela deverá implementar?

2. O que há de errado com a interface seguinte:

```
public interface SomethingIsWrong {  
    void aMethod(int aValue){  
        System.out.println("Hi Mom");  
    }  
}
```

3. Corrija a interface do exercício 2.

4. A interface seguinte é válida?

```
public interface Motor {  
}
```

Exercícios

4. Suponha que você desenvolva um Time Server que envia notificações a aplicações cliente periodicamente (referentes a data e hora atuais). Escreva uma interface que possa ser usada para forçar as aplicações clientes a implementarem seu protocolo de comunicação. Considere que as aplicações cliente devam implementar métodos para definir a hora, definir a data ou definir data e hora ao mesmo tempo. Além disso, as aplicações cliente devem retornar a data e hora a partir do horário local.

5. Escreva uma classe que implementa a interface `java.lang.CharSequence`. Sua implementação deve retornar uma `String` em ordem inversa. Escreva uma aplicação para testar a funcionalidade implementada.