

Programação Orientada a Objetos

Prof. Hugo Marcondes
hugo.marcondes@ifsc.edu.br

Aula 02

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina

Classes e Objetos

² IFSC - Programação Orientada a Objetos

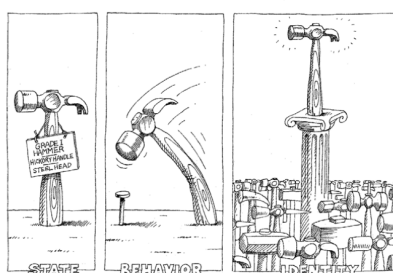
Objetos

- Da perspectiva da cognição humana, um objeto pode ser:
 - Uma coisa tangível/visível
 - Algo que pode ser compreendido intelectualmente
 - Algo em que uma ação pode ser direcionado
- Um objeto representa um indivíduo, uma unidade, item ou entidade identificável, seja real ou abstrata, com um papel bem definido no domínio do problema

³ IFSC - Programação Orientada a Objetos

Objetos

An **object** is an entity that has **state**, **behavior**, and **identity**. The **structure** and **behavior** of **similar objects** are defined in their **common class**. The terms **instance** and **object** are interchangeable.



An object has state, exhibits some well-defined behavior, and has a unique identity.

⁴ IFSC - Programação Orientada a Objetos

Estado de objetos



- O estado de um objeto engloba todas as suas propriedades (geralmente estáticas) além dos valores (geralmente dinâmicos) destas propriedades
- Uma propriedade é uma característica, qualidade que contribui para tornar um objeto único
 - Propriedades são geralmente estáticas
 - Em algumas circunstâncias, uma propriedade de um objeto pode mudar (ex. machine learning)
 - Toda propriedade possui um valor. Este valor pode ser um valor quantitativo, ou pode denotar outro objeto
- O fato de todo objeto possuir um estado, implica que todo objeto ocupa algum espaço (memória)

5 IFSC - Programação Orientada a Objetos

Comportamento de objetos



- Nenhum objeto existe isoladamente.
- Comportamento é como um objeto age e reage, em termos de sua mudança de estado e passagem de mensagens
 - O comportamento de um objeto representa a sua atividade externa visível
- Uma operação é alguma ação que um objeto executa em outro de forma a provocar uma reação
 - Java - métodos
 - C++ - função membro
 - Smalltalk - mensagem
-

6 IFSC - Programação Orientada a Objetos

Comportamento de objetos



- Uma mensagem é simplesmente uma operação que um objeto pode realizar em outro
 - Operação e mensagem são termos intercambiáveis
- A passagem de mensagens é uma parte da equação que define o comportamento de um objeto.
- O estado do objeto também afeta o seu comportamento
 - O estado do objeto representa o resultado cumulativo de seu comportamento

7 IFSC - Programação Orientada a Objetos

Operações de objetos



- Uma operação denota um serviço que é provido para outros objetos
 - **Modificadores:** uma operação que altera o estado de um objeto.
 - **Selecionadores:** uma operação que acessa o estado de um objeto (sem alterá-lo)
 - **Interadores:** uma operação que permite o acesso a "partes" de um objeto em uma ordem bem definida
 - **Construtores:** uma operação para criar e inicializar o estado de um objeto
 - **Destrutores:** uma operação para livrar o estado de um objeto e "destruí-lo".

8 IFSC - Programação Orientada a Objetos

Operações de objetos



- Todos os métodos associados a um objeto particular compreendem o seu protocolo. O protocolo de um objeto define assim o envelope do comportamento permitido de um objeto e assim compreende toda a visão estática e dinâmica do objeto.
- Um papel é uma máscara que um objeto usa e assim define um contrato entre uma abstração e seus clientes
- A existência de estado dentro de um objeto significa que a ordem em que as operações são invocadas é importante
 - Cada objeto é como uma minúscula máquina independente
- Objetos ativos ou passivos. Um objeto ativo é aquele que abrange seu próprio segmento de controle, enquanto que um objeto passivo não

9 IFSC - Programação Orientada a Objetos

Identidade do Objeto



- Identidade é uma propriedade que distingue um objeto de todos os demais
- A identidade única de cada objeto é preservada ao longo do tempo de vida do objeto, mesmo que seu estado seja modificado.

10 IFSC - Programação Orientada a Objetos

Relacionamento entre Objetos



- A relação entre quaisquer dois objetos abrange os pressupostos que cada um faz sobre o outro, incluindo quais operações podem ser executadas e os comportamentos resultantes
- Associações
 - Conexão física ou conceitual entre objetos
 - Denota a associação específica através do qual um objeto (o cliente) se aplica aos serviços de outro objeto (o fornecedor)
- Agregações
 - Enquanto associações denotam uma relação cliente/fornecedor ou "peer-to-peer", agregações denotam a hierarquia todo/parte, com a capacidade de navegar a partir do todo (também chamado o agregado) para as suas partes
 - Agregação é um tipo especializado de associação

11 IFSC - Programação Orientada a Objetos

Relacionamento entre objetos

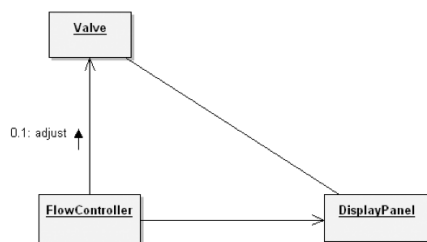
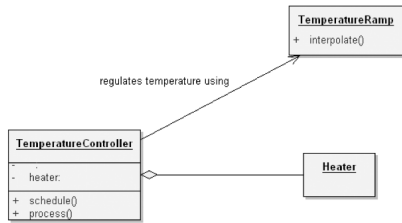


Figure 3-5 Links

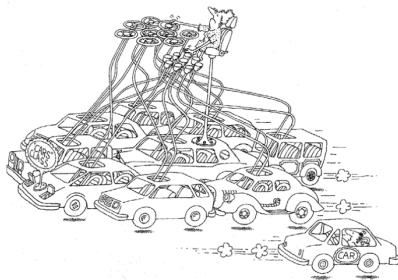
12 IFSC - Programação Orientada a Objetos

Figure 3-6
Aggregation



Classes de Objetos

- Uma classe é um conjunto de objetos que compartilham uma estrutura, um comportamento, e uma semântica comum



A class represents a set of objects that share a common structure and a common behavior.

Interface e Implementação

- A classe serve como um contrato entre abstrações e todos os seus clientes, definido a sua interface.
- A interface de uma classe pode ser dividida em:
 - **Pública:** uma declaração acessível a todos os clientes
 - **Protegida:** uma declaração acessível apenas a própria classe e suas sub-classes
 - **Privada:** uma declaração acessível apenas a própria classe

Interface

- As constantes e variáveis que formam a representação de uma classe é conhecida de diversas formas, de acordo com a linguagem de programação utilizada
 - instance variable (Smalltalk)
 - field (Java)
 - member data (C++)
- O estado de um objeto deve ter alguma correspondência em sua classe, e é tipicamente expressada através de suas constantes e variáveis, declaradas para interface privada ou protegida da classe.
 - Encapsulamento da representação do estado de um objeto!

Relações entre Classes



- Classes, como objetos, não existem isoladamente
 - Abstrações chaves são geralmente relacionadas em variadas formas
- Três tipos básicos de relacionamentos entre classes
 - Generalização / Especialização - “é um(a)”
 - Relação “é parte de”
 - Associação - dependência semântica entre classes

Associações



- É a relação mais genérica e geralmente a mais fraca semanticamente
 - Geralmente expressam dependências entre as abstrações
- Durante a análise é importante capturar tais dependências semânticas, seus papéis e sua cardinalidade

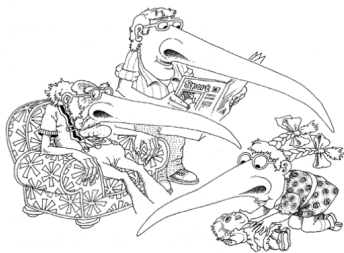


Figure 3-7 Association

Herança



- Herança expressa as relações de generalização e especialização entre classes



A subclass may inherit the structure and behavior of its superclass.

Herança

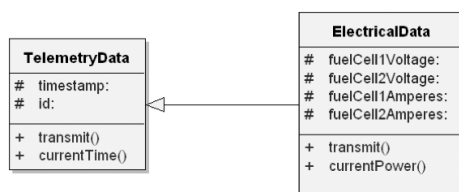


Figure 3-8 ElectricalData Inherits from the Superclass TelemetryData

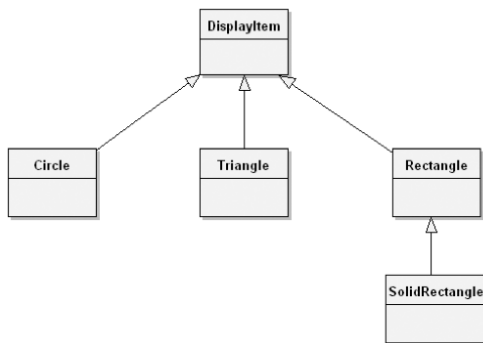
Polimorfismo



- Polimorfismo é um conceito na teoria de tipos, onde um nome pode denotar instâncias de diferentes classes, desde que estas estejam relacionadas por alguma superclasse (herança)
- Qualquer objeto denotado por este nome é capaz de responder a um conjunto de operações de formas distintas
- Polimorfismo é útil quando temos diversas classes com o mesmo protocolo
- Polimorfismo é implementado através do conceito de ligação tardia. Na presença de polimorfismo, a ligação de um método com o seu "nome" não é determinada até a execução do programa.

21 IFSC - Programação Orientada a Objetos

Polimorfismo



22 IFSC - Programação Orientada a Objetos

Agregação



- Relações de agregação denotam a relação de todo-parte.
- Os objetos contidos podem existir sem serem parte do objeto que os contém.



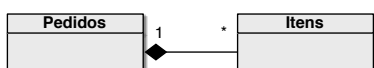
Figure 3-12 Aggregation

23 IFSC - Programação Orientada a Objetos

Composição



- Um tipo de agregação onde objetos contidos não fazem sentido fora do contexto do objeto que os contém.



24 IFSC - Programação Orientada a Objetos

Análise e Projeto



- Durante a análise e dos estágio iniciais do projeto, o desenvolvedor deve:
- Identificar as classes que formam o vocabulário do domínio do problema.
- Definir a estrutura pela qual os objetos trabalham em conjunto para prover o comportamento que satisfaça os requisitos do problema

25 IFSC - Programação Orientada a Objetos

A qualidade de uma Abstração



- Como podemos avaliar se uma classe ou objeto é bem projetado ?
- Acoplamento
 - Dependência entre módulos
- Coesão
 - Grau de conectividade entre elementos de um mesmo módulo
- Suficiência
 - O módulo captura suficientemente as características das abstrações?
- Completude
 - A interface captura todas as características de forma significativa ?
- Operações primitivas
 - Suas operações são primitivas ? Podem ser implementadas eficientemente com a representação da abstração ?

26 IFSC - Programação Orientada a Objetos

Resumindo...



- Um objeto tem estado, comportamento e identidade
- A estrutura e o comportamento de objetos semelhantes são definidos na sua classe comum
- O estado de um objeto engloba todas as propriedades (normalmente estáticas) do objeto mais os valores correntes (usualmente dinâmicas) de cada uma destas propriedades.
- Comportamento é como um objeto age e reage em termos de suas alterações de estado e passagem de mensagens.
- Identidade é a propriedade de um objeto que o distingue de todos os outros objetos.

27 IFSC - Programação Orientada a Objetos

Resumindo...



- Uma classe é um conjunto de objetos que compartilham uma estrutura comum e um comportamento comum
- Os três tipos de relacionamentos incluem associação, herança e agregação
- Abstrações-chave são as classes e objetos que formam o vocabulário do domínio do problema
- Um mecanismo é uma estrutura em que um conjunto de objetos trabalham juntos para fornecer um comportamento que satisfaça alguma exigência do problema
- A qualidade de uma abstração pode ser medida por seu acoplamento, coesão, suficiência, completude e primitivismo

28 IFSC - Programação Orientada a Objetos
