INSTITUTO FEDERAL
Santa Catarina

Técnico em Desenvolvimento de Sistemas

Projeto de Software I

Juliano Lucas Gonçalves

juliano.goncalves@ifsc.edu.br / julianolg@gmail.com

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Agenda

- Porque Utilizar Orientação a Objetos?
- Conceitos
- Análise Orientada a Objetos (OOA)

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Por que a Orientação a Objetos?

- As abstrações podem corresponder às “coisas” do domínio do problema, facilitando o entendimento
- Esta abstração facilita a comunicação com os usuários
- Os mesmos objetos existem em todas as fases e uma notação única facilita a INTEGRAÇÃO ENTRE FASES de desenvolvimento
- A tecnologia de objetos facilita o entendimento do domínio do problema, permitindo o GERENCIAMENTO DA COMPLEXIDADE através da modularização
- Facilidade de mudanças através do ENCAPSULAMENTO de dados

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Conceitos Envolvidos

- OBJETO
- MÉTODO
- MENSAGEM
- CLASSE
- GENERALIZAÇÃO
- ESPECIALIZAÇÃO
- HERANÇA
- POLIMORFISMO
- SOBRECARGA
- ENCAPSULAMENTO
- ABSTRAÇÃO
- MODULARIZAÇÃO

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

OBJETO

- Entidades que possuem **dados e instruções** sobre como manipular estes dados
- Os objetos estão ligado à solução do problema.
  - Software Gráfico – Objetos: Círculos; Linhas; etc.
  - Software BD – Objetos: Tabelas; Linhas; Campos; etc.
  - Software Comercial: Pedidos; Produtos; Clientes; etc.
- Na OO a solução do problema consiste em um primeiro momento estabelecer quais os objetos serão necessários.

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

MÉTODOS

- Métodos são procedimentos que determinam como o objeto se comporta.
- Através dos métodos é possível manipular os dados contidos no objeto.
- Os métodos estão ligados ao comportamento do objeto
- Exemplos
  - Um círculo poderia possuir os métodos:

draw; move; getArea; getPerimeter; setCenter

  - Um cliente poderia possuir os métodos:

calculaIdade; getTelefone

  - Um Telefone poderia possui os métodos:

tocar; discar

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

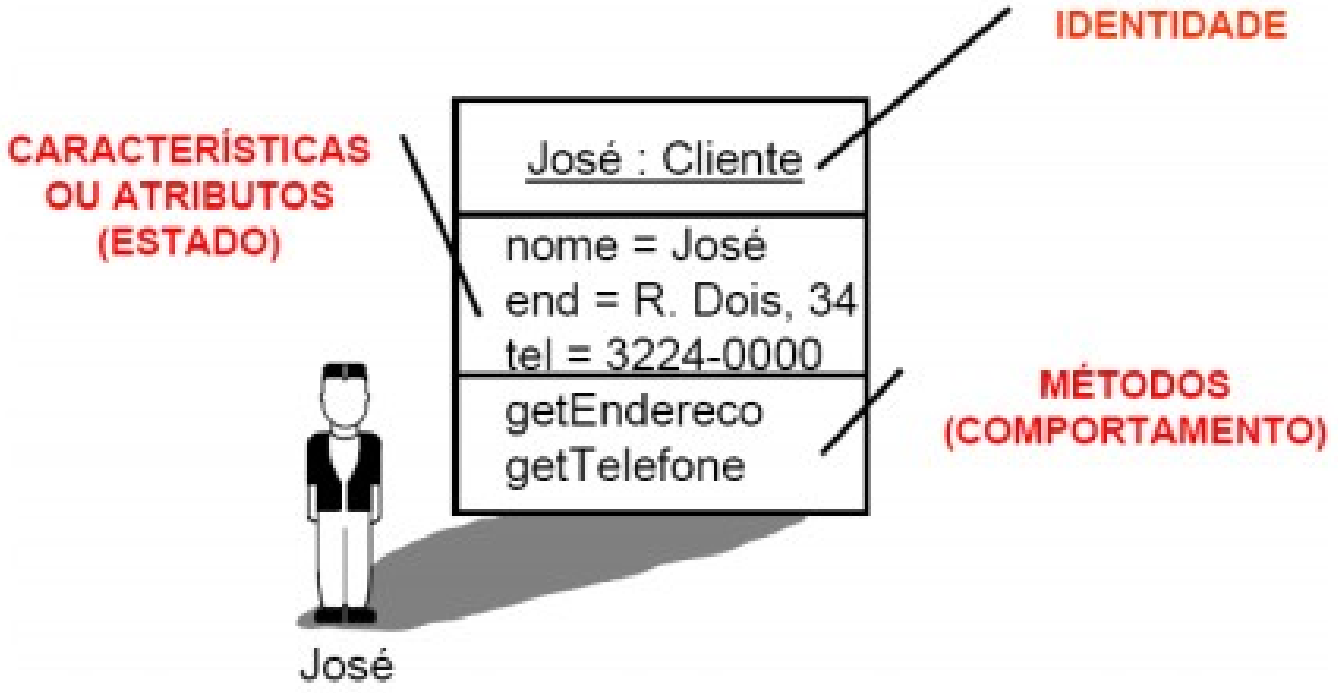
MÉTODOS

- Um método é a implementação de uma operação
- Métodos só tem acesso aos dados da classe para a qual foram definidos
- Métodos normalmente possuem argumentos, variáveis locais, valor de retorno, etc.
- Alguns métodos especiais:
  - Construtores – Criam objetos
  - Destrutores – Destroem objetos
  - Acessores – Recuperam o estado de um atributo (getNomeAtributo)
  - Modificadores – Alteram o estado de um atributo (setNomeAtributo)

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

OBJETOS - RESUMO



INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

MENSAGEM

- Objetos se comunicam entre si através de mensagens.
- Uma mensagem é uma chamada de um método.
- A mensagem possui os seguintes componentes:
  - Receptor – nome do objeto que irá receber a mensagem
  - Método – Método do receptor que será utilizado
  - Argumentos – Informação adicional para a execução do método
  - **Exemplos**

```
Point p(0,0), pNewCenter(2,3);
Circle c(p,3);
c.getArea(); //Exemplo Mensagem
c.setCenter(pNewCenter); //Exemplo Mensagem
```

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Generalização e Especialização

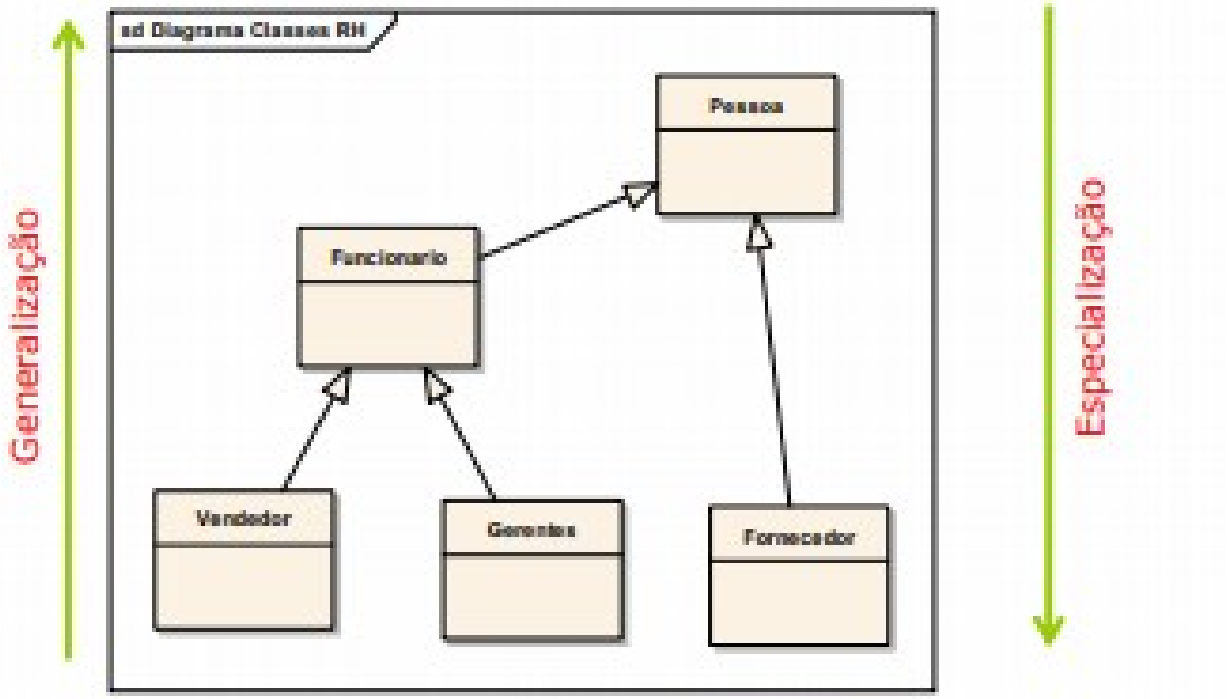
- GENERALIZAÇÃO
  - A generalização consiste em obter similaridades entre as várias classes e partir destas similaridades, novas classes são definidas.
  - Estas classes são chamadas **superclasses**
- ESPECIALIZAÇÃO
  - A especialização por sua vez consiste em observar diferenças entre os objetos de uma mesma classe e dessa forma novas classes são criadas.
  - Estas classes são chamadas **subclasses.**

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Generalização e Especialização Exemplo

- Hierarquia de classes



INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

HERANÇA

- Herança é a capacidade de uma subclasse de ter acesso as propriedades da superclasse a ela relacionada.
- Dessa forma as propriedades de uma classe são propagadas de cima para baixo em um diagrama de classes.
- Neste caso dizemos que a subclasse herda as propriedades e métodos da superclasse
- A relação de herança entre duas classes é uma relação da seguinte forma: A "e um tipo de" B, onde A e B são classes.

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

HERANÇA

- Exemplos:
  - Um Carro de Passeio "é um tipo de " veículo; Um caminhão "é um tipo de" veículo;
  - Um círculo "é um tipo de" uma figura geométrica; Um quadrado "é um tipo de" figura geométrica;
  - Um vendedor "é um tipo de" Empregado; Um empregado "é um tipo de" pessoa.
- Herança Múltipla
  - Uma subclasse herda características de mais uma classe
  - Exemplos
    - Um gerente de vendas "é um tipo" de vendedor e também "é um tipo de" gerente;

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

HERANÇA x USO

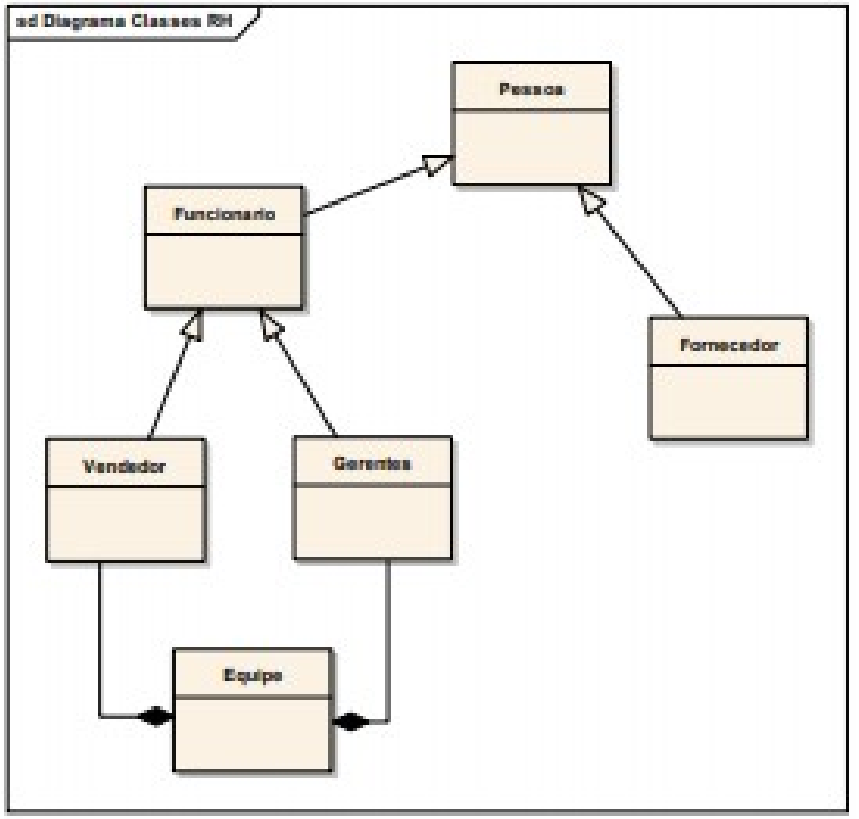
- Além da relação de herança entre as classes existe a relação de uso
- HERANÇA
  - classe A "é um tipo de" B
- USO / AGREGAÇÃO (Relação de Conteúdo)
  - classe D "contém" classe C"
  - classe D "usa" classe C"
  - classe C "é parte da" classe D
  - Exemplo: Uma **equipe contém um gerente e** um grupo de **vendedores**

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Herança x Uso

- Exemplo



INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

POLIMORFISMO (Override)

- Os objetos respondem às mensagens que eles recebem através dos métodos.
- A mesma mensagem pode resultar em diferentes resultados. Esta propriedade é chamada de **polimorfismo**
  - **Exemplo: Método getSalario()**
    - Para um empregado qualquer → getsalario() = Salario;
    - Para o gerente → getsalario() = salario + bonificacao;
  - **Exemplo: Método draw()**
    - Para uma figura qualquer desenha uma forma não definida
    - Para o retângulo, triângulo e círculo o mesmo método responde de uma forma diferente

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

SOBRECARGA(Overload)

- Nas linguagens orientadas a objetos é possível, em uma classe, a existência de métodos que possuem o mesmo nome, porém com diferentes assinaturas.
- Este conceito é chamado de Sobrecarga (Overload)
  - **Exemplo: Em uma classe Figura, temos os seguintes métodos**
    - draw() → desenha o objeto e utiliza uma cor padrão
    - draw(Color) → desenha o objeto, porém recebe uma cor como parâmetro

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

ENCAPSULAMENTO

- Conceito que indica que os dados contidos em um objeto somente poderão ser acessados e/ou modificados através de seus métodos.
- Dessa forma não é possível alterar os dados diretamente, somente através de métodos definidos no objeto
- Exemplo
  - O raio somente pode ser alterado/recuperado pelos métodos setCenter/getCenter.

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

ABSTRAÇÃO

- Abstração é o processo de identificar as qualidades ou propriedades importantes do problema que está sendo modelado.
- Através de um modelo abstrato,pode-se concentrar nas características relevantes e ignorar as irrelevantes.
- Abstração é fruto do raciocínio.
- Através da abstração é possível controlar a complexidade. Isto é feito através da ênfase em características essenciais, fazendo-se uma supressão daquilo que não está ligado ao domínio do problema.

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

MODULARIZAÇÃO

- Consiste em decompor o problema em partes menores.
- Dessa forma o foco é mantido em itens(classes; pacotes; etc.) menores, coesos e fracamente acoplados.

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Análise Orientada a Objetos (OOA)
Projeto Orientado a Objetos (OOD)

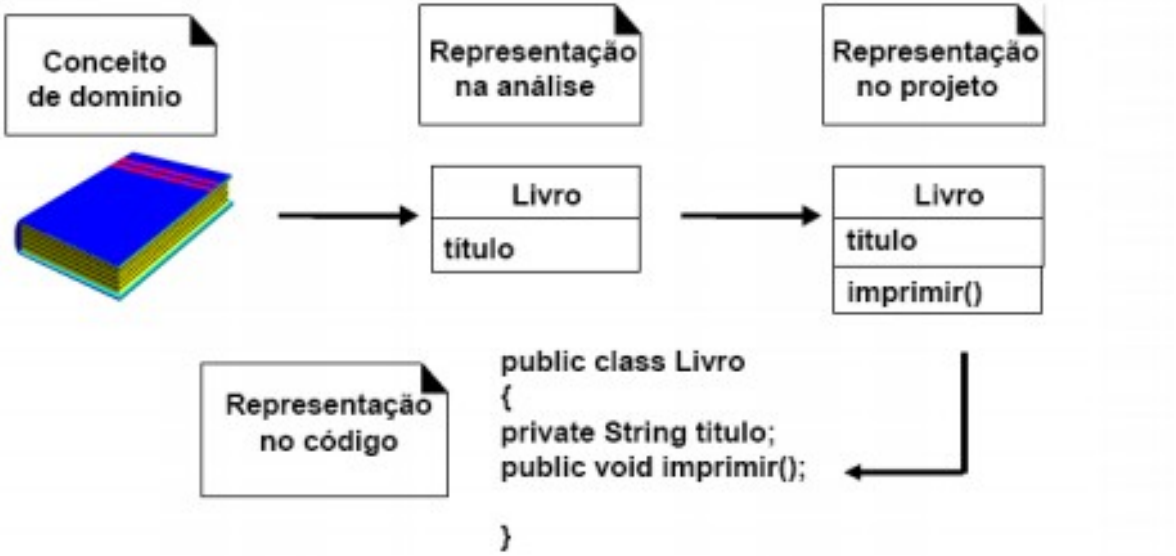
- As técnicas tradicionais (Análise e Projeto Estruturado) não são adequadas para o desenvolvimento de software utilizando a orientação à objetos.
- A utilização do orientação à Objetos solicita uma nova forma de abstrair e entender o problema.
- A linguagem UML é um padrão de diagramação para visualizar os resultados da análise e projeto orientados à objetos

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Análise Orientada a Objetos (OOA)
Projeto Orientado a Objetos (OOD)

- Exemplo
  - O conceito "Livro" em um sistema de biblioteca



INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Análise Orientada a Objetos(OOA)

- Objetivo básico
  - Identificar classes a partir das quais objetos serão representados como instâncias
- Envolve as seguintes tarefas
  - Identificação de Objetos
  - Especificação de Atributos
  - Definição de métodos
  - Comunicações entre objetos

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Análise Orientada a Objetos(OOA)

- IDENTIFICAÇÃO DE OBJETOS
  - Entidades externas (Outros sistemas; dispositivos;Pessoas)
  - Coisas ligadas ao domínio do problema (Relatórios;Displays;...)
  - Ocorrências ou Eventos (Conclusão de um movimento;Alarme disparado; Clique do mouse; etc.)
  - Papéis ou funções (Engenheiro; Gerente; Vendedor) desempenhados por pessoas
  - Unidades organizacionais (Grupo; Equipe;...)
  - Lugares (Piso de fábrica; área de descarga)
  - Estruturas (Sensores; veículos de quatro rodas;...)

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

Análise Orientada a Objetos(OOA)

- Em uma especificação:
  - NOMES são potenciais objetos (e classes)
  - VERBOS são potenciais métodos
- A regra acima deve ser utilizada apenas como referência.
- O entendimento do contexto, das necessidades do usuário são fundamentais para classificar possíveis objetos e métodos

INSTITUTO FEDERAL
Santa Catarina

Projeto de Software I

- Importante

  Esses conceitos serão vistos na disciplina de Programação Orientada a Objetos na fase 3