Campanha Nacional de Escolas da Comunidade
Faculdade Cenecista Nossa Senhora dos Anjos
Credenciada pela Portaria Ministerial nº 1.267/2002 - D.O.U de 26/04/2002

Autor: Daniel Wildt (dwildt@gmail.com)

Resumo sobre Orientação a Objetos

Classe
Representação, abstração do mundo real. Exemplo: Carro, Computador, Caneta, Pessoa. É a representação de uma entidade do sistema real em um modelo computacional

Objeto
Abstração que se torna realidade. Caneta -> BIC, Carro -> Fusca, Pessoa -> José Silva

Atributos e operações
Para cada representação que definimos, podemos informar atributos (variáveis) e operações (métodos ou ações) que esta representação possui e executa.

Abstração
Trazer a representação de uma entidade do mundo real para o mundo computacional

Encapsulação
Ocultar as especificades de funcionalidades.

Herança
Conceitos onde uma classe é a base de outra.
Programação por especialização.
Conceitos de generalização / especialização.

Associação
Associação existente entre duas entidades.
Cliente possui Pedidos. Um pedido é referente a um cliente.

Agregação
Considerar algo maior pensando na relação todo-parte.
Um Carro C possui um Motor M. Se o carro for removido do sistema, o motor que estava sendo desenvolvido pode ser reaproveitado em outro carro.

Composição
Também baseado na relação todo-parte.

Neste caso, se o objeto maior for removido, as suas partes filhas serão removidas também.
Imagine o caso de um Carro C que possui uma Placa P registrada. Se o carro deixa de existir, a placa não tem mais utilidade dentro do sistema.

Pensamentos em exemplos
Herança: Pessoa física "é um tipo" de Pessoa.
Associação: Um pedido "usa um" cliente dentro da sua estrutura.
Agregação: Um carro "é composto" de um Motor, etc.

Polimorfismo
Capacidade de objetos derivados de uma mesma base reagirem de forma diferente.
Classe Animal. Pensar em operação de andar, onde um Cachorro e uma Pessoa andam de forma diferente, mas são derivados de uma mesma base: Animal.

Classes Abstratas
Classes base para outras classes.
Alguns métodos não possuem implementação (filhos devem implementar).

Visibilidade (privado, público, protegido)
Público (public): operação e/ou atributo pode ser acessado de outras entidades.
Protegido (protected): operação e/ou atributo pode ser acessado pela entidade onde foi definido e pelos filhos.
Privado (private): operação e/ou atributo pode acessado apenas pela entidade onde foi definido.

Interfaces
Contratos que classes podem assinar.
Imagine a situação:
- Sistema possui Classes Professor, Aluno.
- Cliente quer permitir que um professor seja aluno.
- Como fazer isto no sistema, se muitas linguagens não possuem conceitos de herança múltipla?
- Solução: Contratos!

Solução:
- Criar um contrato com as operações que um aluno implementa.
- Criar uma classe derivada de Professor que assina o contrato de aluno.
- Desta forma se consegue usar conceitos de polimorfismo para resolver o problema.

Mais reuso de código
Maior padronização

Referências de apoio
http://ootips.org