

Aula 13 - Engenharia de Requisitos

Classes e Objetos - Orientação a Objeto

É uma forma de entender e representar sistemas complexos como estruturas hierárquicas de objetos correlatos.

Objetos:

- Um objeto é uma construção de software que encapsula estado e comportamento, através de propriedades (atributos) e operações (métodos);
- Estado de um objeto: composto por suas propriedades e seus respectivos valores
- Comportamento: a maneira como o objeto reage quando o seu estado é alterado ou quando uma mensagem é recebida

Classes:

- Conjunto de objetos similares
- Um objeto é uma instância de uma classe
- Objetos de uma mesma classe diferenciam-se pelos valores de suas propriedades e de seus identificadores únicos.

Troca de mensagens:

- Mecanismo através do qual os objetos se comunicam, invocando as operações desejadas
- Um objeto (Emissor) envia uma mensagem a outro (Receptor) que executará uma tarefa
- Operações (métodos) são invocados através de mensagens

Encapsulamento:

- Objetivo: Ocultar do mundo externo ao objeto, os detalhes de implementação e restringir o acesso às propriedades e métodos.
- Vantagens
 - Segurança no acesso ao objeto
 - Melhor consistência no estado interno, pois evita alterações incorretas de valores das propriedades
- Uma maneira de visualizar uma classe é a que consiste de duas partes: a interface e a implementação (interface - vista e usada por outros objetos e implementação - escondida dos outros objetos)

Operação:

- Uma classe incorpora um conjunto de responsabilidades que definem o comportamento dos objetos na classe (as responsabilidades de uma classe são executadas por suas operações)
- Não é necessariamente um mapeamento um-para-um

- Exemplo
 - Classe Produto - Responsabilidade de fornecer preço
 - Operações: buscar informações no banco, calcular preço, etc.

Atributos:

- Um atributo é uma característica de uma classe
- Atributos não tem comportamento (eles não são objetos)
- Nomes de atributos são substantivos simples ou frases substantivas

Classes Entidade

- Uma classe entidade modela informação e comportamento associado que geralmente tem uma vida longa
- Este tipo de classe pode refletir uma entidade do mundo real, ou ela pode ser necessária para executar uma tarefa interna do sistema
- Muitas vezes, elas são aplicações independentes, isso significa que muitas delas poderão ser utilizadas por mais de uma sistema
- As classes entidades são normalmente utilizadas pelo sistema para definir alguma responsabilidade. Os nomes e as frases usadas para descrever a responsabilidade podem ser um bom ponto de partida.
- Frequentemente chamadas de classes de domínio, desde que elas usualmente tratam com abstrações de entidades do mundo real

Classes Limite

- Classes Limite cuidam da comunicação entre meio com o qual o sistema interage e o sistema propriamente dito.
- Fornecem a interface para um usuário ou para um outro sistema e são utilizadas para modelar as interfaces do sistema
- Classes Limite são também adicionadas para facilitar a comunicação com outros sistemas. Durante o projeto, estas classes são refinadas para levar em consideração o protocolo de comunicação escolhido.

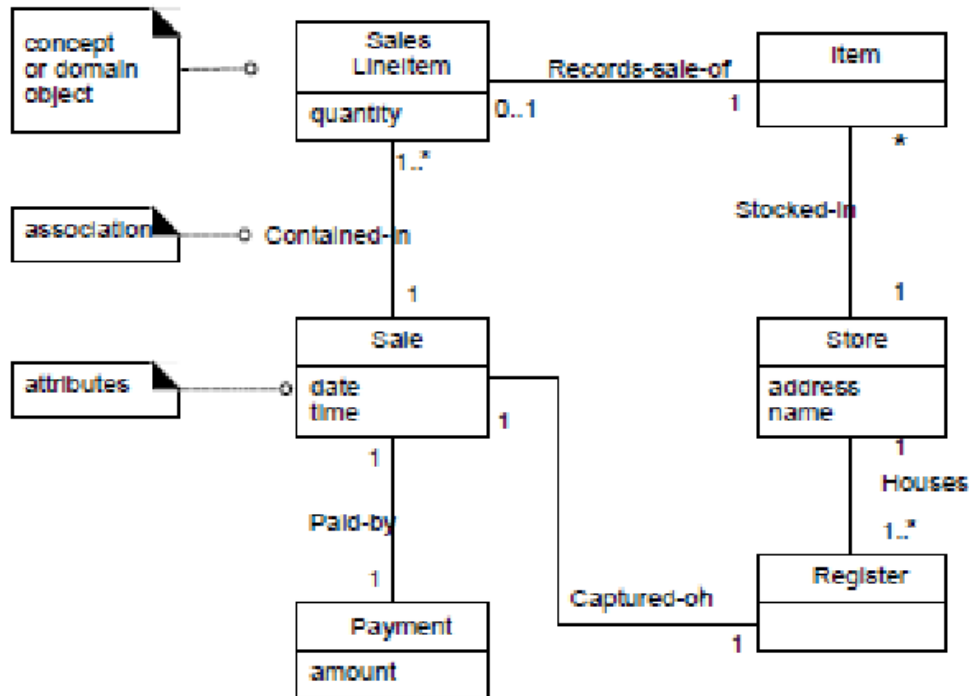
Classes Controle

- Classes Controle modelam uma sequência de comportamento específico a um ou mais casos de uso.
- Coordenam os eventos necessários para a realização do comportamento especificado em um caso de uso, sendo dependentes da aplicação
- Se uma Classe Controle está fazendo mais do estabelecer uma sequência, então ela está fazendo coisas demais.

Modelo de Classes Conceituais

- Um modelo de domínio ilustra classes conceituais significativas (para os modeladores) em um domínio de problema

- Identificar um rico conjunto de objetos ou classes conceituais é o cerne da análise orientada a objetos e vale o esforço em termos de recompensa durante o trabalho de design e implementação.
- Um modelo de domínio é uma representação de classes conceituais do mundo real, não de componentes de software. Não é um conjunto de diagramas que descrevem classes de software, ou objetos de software com responsabilidades.
- Exemplo



- Usando a notação UML, um modelo de domínio é ilustrado com um conjunto de diagramas de classes em que nenhuma operação é definida. Pode mostrar:
 - objetos de domínio ou classes conceituais
 - associações entre classes conceituais
 - atributos de classes conceituais
- Classes Conceituais
 - Informalmente, uma classe conceitual é uma ideia, coisa ou objeto. Mais formalmente, um conceito classe pode ser considerado em termos de seu símbolo, intenção e extensão.
 - Símbolo - palavras ou imagens que representam uma classe conceitual
 - Intenção - a definição de uma classe conceitual
 - Extensão - o conjunto de exemplos aos quais a classe conceitual se aplica
 - A tarefa central é, portanto, identificar as classes conceituais relacionadas aos cenários em projeto.
 - É melhor especificar demais um modelo de domínio com muitos conceitos conceituais de baixa granularidade do que para sub especificá-lo.
 - Estratégias para identificar classes conceituais
 - Use uma lista de categorias de classe conceitual

- Identifique Frases nominais