Aula 5

Análise de Sistemas

Profª Adriana Bastos da Costa

Conversa Inicial

Análise de sistemas

- Projetar e construir um programa de computador não é uma tarefa trivial, pois envolve conhecer bem o problema que se quer resolver e traduzir a solução ideal para o problema em linhas de código, portanto, entender bem os requisitos ou o que o software precisa fazer para atender às necessidades do cliente é base de tudo

Análise de sistemas

- Se os requisitos não forem bem analisados e o software não for projetado de maneira adequada, fica mais difícil projetar uma solução que realmente resolva o problema do cliente

Diagrama de classes

- Nesta aula, vamos conversar sobre um diagrama muito importante para o projeto de software, que é o diagrama de classes. Vamos também compreender como é estruturado e qual é o objetivo do diagrama de classes

Modelagem de processos de negócio

- Esta aula estará organizada em 5 grandes temas
  - Entendendo o diagrama de classes
  - Classes, atributos e métodos
  - Relacionamentos
  - Técnicas de modelagem de diagrama de classes
  - Analisando um exemplo de diagrama de caso de uso

Entendendo o diagrama de classes

Entendendo o diagrama de classes

- Os diagramas de classes são fundamentais para o processo de modelagem de objetos e modelam a estrutura estática de um sistema. Dependendo da complexidade de um sistema, é possível utilizar um único diagrama de classes para modelar um sistema inteiro ou vários diagramas de classes para modelar os componentes de um sistema

Orientação a objetos

- A modelagem das classes está totalmente relacionada com os conceitos de orientação a objetos. Entre os principais conceitos da orientação a objetos, já estudados na aula 4, vamos relembrar
- Abstração: tem o foco em aspectos relevantes para um determinado propósito, abstraindo os demais elementos que não são importantes para a situação que se está modelando

Orientação a objetos

- Encapsulamento: consiste na separação dos aspectos externos de um objeto, acessíveis por outros objetos, dos detalhes internos da implementação daquele objeto, que ficam ocultos dos demais objetos
- Herança: é o compartilhamento de atributos e operações entre classes com base em um relacionamento hierárquico. Permite que uma classe herde características de outra classe

Diagrama de classe

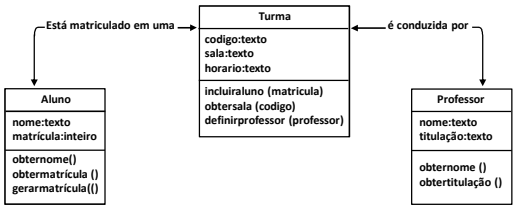


Diagrama de classe

- O diagrama de classes modela o entendimento sobre quais são as classes relacionadas com o escopo do software, assim como o relacionamento entre elas, mostrando como as informações se encaixam e se comunicam entre si para que o software faça tudo o que precisa ser feito para atender às necessidades do cliente

Classe X Objeto

- Imagine que temos uma classe ALUNO e, dentro dessa classe, temos uma instância para cada aluno da escola, ou seja, "José da Silva" é um objeto da classe ALUNO, assim como a "Maria dos Santos" é outro objeto da classe ALUNO

Classe X Objeto

- Dessa forma, seguindo os conceitos da orientação a objetos, toda vez que crio um novo objeto dentro da classe dizemos que criamos uma nova instância da classe ou a instanciamos

Classe

- As classes podem ser entendidas como uma abstração de um conjunto de coisas que possuem características e operações em comum
- O domínio do problema são as informações referentes à classe identificada para o escopo do software que está sendo modelado e será construído

Atributos

- Os atributos representam o conjunto de características ou estados dos objetos de uma determinada classe. Os atributos podem também ser entendidos como propriedades semelhantes que os objetos de uma classe possuem

Atributos

- Cada atributo permite definir um intervalo de valores que as instâncias dessa propriedade podem apresentar
- O aluno se chama "João da Silva" e tem a matrícula "123456789". Essas propriedades de aluno são descritas pelos atributos nome e matrícula, na classe ALUNO

Métodos

- Os métodos representam o conjunto de operações ou comportamento que a classe fornece ao software ou que a classe é responsável por executar
- Por exemplo, em uma classe ALUNO, é preciso executar ações como consultar aluno, alterar dados de aluno, entre outras. Essas ações, no diagrama de classes, seguindo a UML, são chamadas métodos

Funcionamento do método

- Um método pode ou não retornar um valor
- Um método pode ou não aceitar argumentos
- Um método, após encerrar sua execução, retorna o fluxo de controle do programa para quem o chamou

Características dos atributos e métodos

- Definido como + público, é visível em qualquer classe de qualquer pacote
- Definido como # protegido, é visível para classes do mesmo pacote
- Definido como – privado, é visível somente para a classe onde foi definido

Relacionamentos

Relacionamentos

- O objetivo dos relacionamentos entre as classes é garantir a comunicação e o compartilhamento de informações entre elas, mostrando em detalhes como ocorre a colaboração de umas com as outras

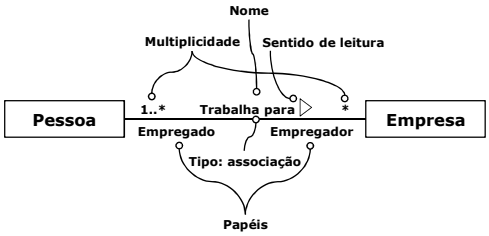
Características do relacionamento

- Nome: descrição dada ao relacionamento
- Sentido de leitura: mostra qual classe é a origem e qual classe é o destino do relacionamento
- Navegabilidade: é indicada por uma seta no fim do relacionamento e está relacionada com o sentido da leitura que será feito para compreender o relacionamento

Características do relacionamento

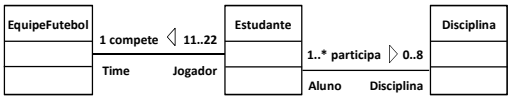
- **Multiplicidade: indica como o relacionamento poderá ser verificado na classe e é muito importante para entender as regras de negócio de um requisito do software**
- **Tipo: representa o tipo do relacionamento modelado de uma classe com a outra**
- **Papéis: desempenhados pela classe em um relacionamento**

Características do relacionamento



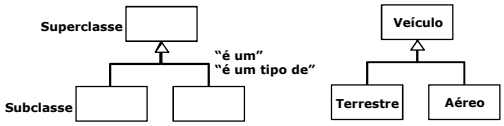
Tipos de relacionamentos

- **Associação – é um relacionamento estrutural que indica que os objetos de uma classe estão vinculados a objetos de outra**



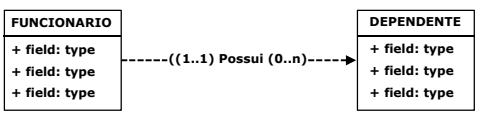
Tipos de relacionamentos

- **Generalização – é um relacionamento que utiliza o conceito de herança, da orientação a objetos, permitindo que as subclasses herdem características da superclasse**



Tipos de relacionamentos

- **Dependência – representa a dependência entre classes quando estas sofrem mudanças de estado. Com esse relacionamento, é possível entender o impacto e como uma classe é afetada quando a outa classe é modificada**



Técnicas de modelagem de diagrama de classes

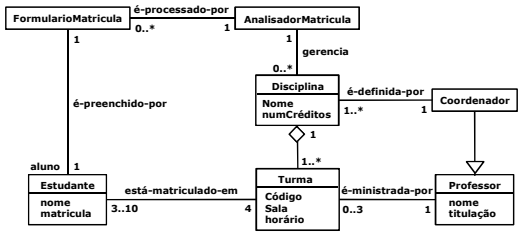
Modelando um diagrama de classe

- Fazer a análise de um software não é algo trivial e envolve uma lógica complexa para transformar as necessidades de um cliente, que é algo ainda intangível e abstrato, em algo mais objetivo, que ajude a compreender como o software deverá ser implementado para gerar uma solução eficiente para resolver, da melhor forma possível, o problema que o cliente precisa resolver com a automatização e construção de um software

Passos a serem seguidos

- Levantar os requisitos
- Modelar o diagrama de caso de uso
- Descrever os casos de uso
- Identificar as classes
- Identificar os atributos e métodos
- Modelar o diagrama de classes

Diagrama de Classes



Analisando um exemplo de diagrama de caso de uso

Estudo de caso

- O nosso estudo de caso será o seguinte: "fomos contratados pelo nosso cliente para modelar o processo de vendas online de livros. O nosso cliente tem uma livraria virtual, que vende produtos diretamente em um site próprio

Estudo de caso

- O diferencial desta livraria é ter um estoque próprio, o que garante uma entrega mais rápida a seus clientes, e aceitar vários tipos de pagamento, como cartão de crédito, cartão de débito e boleto bancário
- A livraria possui um programa de fidelidade, que permite desconto de 10% aos clientes que comprarem R$ 500,00 ou mais em 1 ano"

UC01 - Utilizar programa de fidelidade

- O UC01 é chamado após o cliente selecionar um ou mais produtos e clicar em carrinho de compras, na tela de compras de produto. Este caso de uso vai mostrar, na tela do carrinho de compras o valor de desconto aplicado à compra, caso o cliente seja elegível ao desconto, por ter atingido R$ 500,00 em compras realizadas no último ano

UC01 - Utilizar programa de fidelidade

- Se o cliente não tiver atingido o valor de compras que o habilita para o desconto de fidelidade, o campo de desconto será apresentado na tela do carrinho de compras com R$ 0,00

UC02 - Atualizar programa de fidelidade

- O UC02 é um caso de uso interno do sistema, sem interação com o usuário, por isso é um caso de uso sem MSG. O objetivo do caso de uso é identificar se um cliente é elegível a ter o desconto por fidelidade na próxima compra que efetuar

Classes identificadas

- A partir do entendimento sobre os casos de uso, é possível listar as seguintes classes principais: CLIENTE, FIDELIDADE, PEDIDO e PRODUTO DO PEDIDO
- Na classe CLIENTE, podemos listar os principais atributos sendo: Nome (texto), CPF (inteiro), Telefone (inteiro) e Endereço (texto)

Classes identificadas

- Na classe FIDELIDADE, os principais atributos são: Status Fidelidade (boolean), que diz se o cliente tem ou não direito à fidelidade, e o atributo CPF (inteiro) para identificar a qual cliente se refere

Classes identificadas

- Na classe PEDIDO, podemos listar os atributos: NÚMERO (inteiro), para identificar o pedido em si; CPF (inteiro), para relacionar a qual cliente se refere; DATA DO PEDIDO (data), mostrando o dia, mês e ano da realização do pedido. VALOR DO PEDIDO (real), mostrando o valor total gasto pelo cliente no pedido

Classes identificadas

- **Foi necessário criar uma classe de PRODUTO DO PEDIDO para evitar repetir os dados do cliente e do pedido em si, em cada produto comprado pelo cliente no mesmo pedido**

Classes identificadas

- **Os principais atributos dessa classe são: NÚMERO (inteiro), que é o atributo que identifica a qual pedido os produtos da classe estão relacionados; PRODUTO (inteiro), que é um atributo para identificar o produto adquirido no pedido em questão; QUANTIDADE PRODUTO (inteiro), que identifica qual a quantidade do produto em questão foi adquirida pelo cliente; VALOR UNITÁRIO (real), que é o valor unitário do produto adquirido**