

Aula 08

Padrão de projeto orientado a objetos

Projeto Geral:

traduz a especificação do sistema em termos da arquitetura de dados e de módulos. Principal envolvido: arquiteto

Projeto Detalhado: refinamento visando à codificação e especificação dos programas. Principal envolvido: arquiteto e desenvolvedores

Modularização : o tipo da modularização define o tipo de projeto.

O sistema será particionado em um conjunto de objetos que se comunicam para resolver um determinado problema.

Análise OO

A ênfase está em achar e descrever objetos (ou conceitos) no domínio do problema

Projeto OO

Durante o projeto OO, a ênfase está em achar objetos lógicos de software que poderão ser eventualmente implementados usando uma linguagem OO

Atividades do projeto OO

- Atribuição de responsabilidades entre os objetos
- Construção de diagramas de classes - ESTRUTURA
- Construção de diagramas de interação (seqüência e colaboração) - COMPORTAMENTO
- Levantamento de necessidades de concorrência
- Considerações de tratamento de defeitos
- Detalhamento do formato de saída (interface com usuário, relatórios, transações enviadas para outros sistemas, ...)
- Definição do esquema do BD: mapeamento de objetos para tabelas se o BD for relacional

Elementos da análise que serão mapeados para o projeto OO

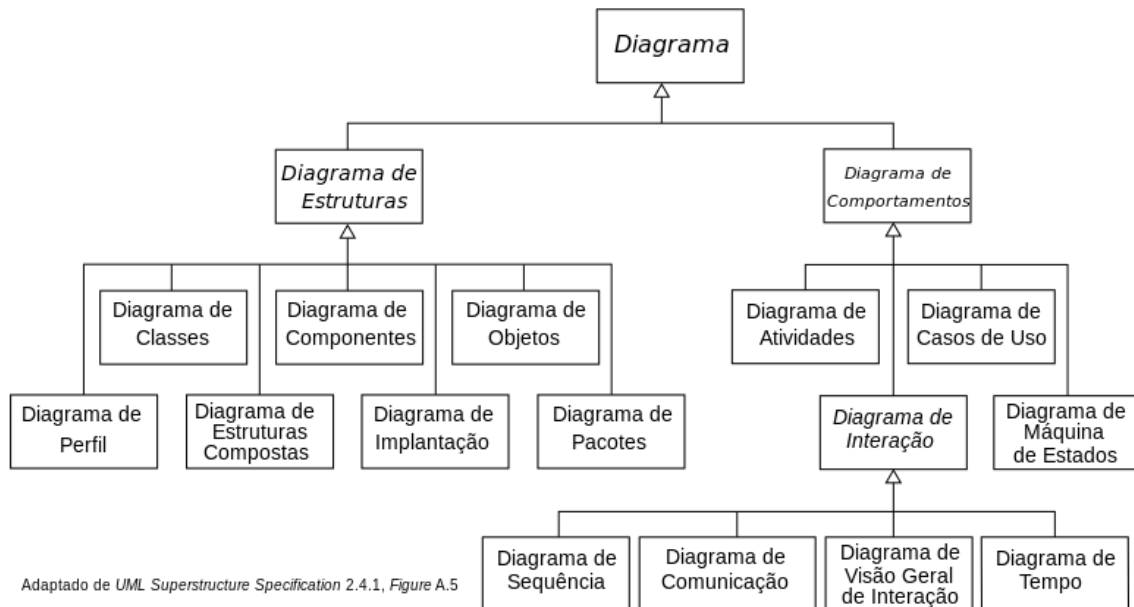
- Casos de uso → processo de domínio
- Modelo conceitual → conceitos e termos
- Diagrama de seqüência de eventos → eventos e operações
- Contratos → o que fazem as operações
- Cartões CRC → classes, suas responsabilidades e colaborações

A partir dos artefatos da fase de análise é desenvolvida uma solução lógica. Os **Diagramas de Interação (Seqüência e/ou Colaboração)** são a base de tal solução, a partir deles, posteriormente, são criados os Diagramas de Classes (de projeto). **E O INVERSO ??**

Padrão UML → Unified Modeling Language

Padrão atualmente gerido pela OMG - Object Management Group, com diversos diagramas para desenvolvimento de software, desde a fase de análise de requisitos, até teste e deployment.

Um processo típico – RUP → Rational Unified Process → derivado em Unified Process



Diagramas de classe

Um dos diagramas UML mais usados, para fazer representação de estruturas das classes do negócio, e também de requisitos não funcionais. Serve de apoio para construção da maioria dos demais diagramas, é considerado diagrama central para desenvolvimento usando UML, e até mesmo para outras notações.

Um diagrama UML pode ser realizado manualmente, ou através de muitas ferramentas existentes, com suporte a diferentes capacidades.

Um diagrama de classes possui notação específica para representar diferentes conceitos de projetos orientados a objeto: classes, atributos, mensagens, encapsulamento, operações e seus parâmetros.

As regras para descoberta e especificação de modelos conceituais podem em grande parte mapeados para diagramas de classes, com elementos adicionais relativos à implementação.

Ferramentas (online e offline):

Uma ferramenta deve ser escolhida seguindo critérios específicos de cada projeto, não é possível afirmar que uma ferramenta é a “*melhor*” de todas. A escolha é feita pelo(s) arquiteto(s) do projeto.

<http://draw.io> - imagem, pdf

<https://app.genmymodel.com/> , imagem, pdf, xmi

<http://staruml.io> – boa interface, mas não gera xmi

[Argouml.tigris.org](http://argouml.tigris.org) – fácil de lançar, mas antiga. Gera xmi.

Papyrus Eclipse – gera xmi

Eclipse CHE – Online. Interessante para pequenos testes. Mas muito lento.

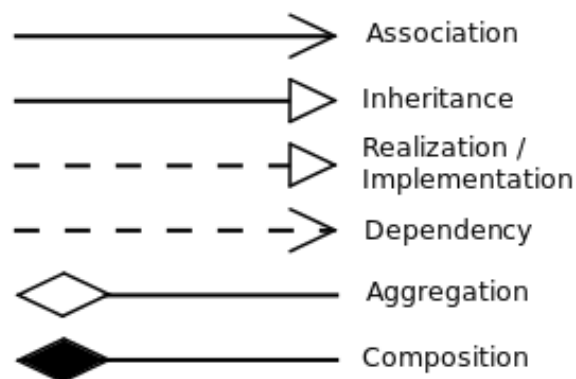
<https://modeling-languages.com/web-based-modeling-tools-uml-er-bpmn/>

<https://modeling-languages.com/uml-tools/#textual>

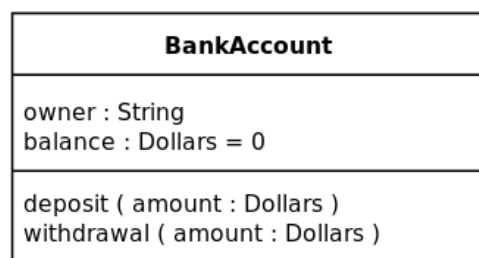
<http://yuml.me/>

Creately

Diagrama de redes sociais como exemplo ilustrativo. Cada elemento visual da notação tem um significado, para representar um modelo de classes OO.



Estrutura da classe



1 MEMBROS

- As classes são formadas por atributos e métodos, que são **membros da classe**.
- Os atributos possuem tipo, e os métodos possuem tipo de retorno, parâmetros e seus tipos

Os **membros** possuem visibilidades:

- + → pública
- → privada
- # → protegida
- ~ → pacote

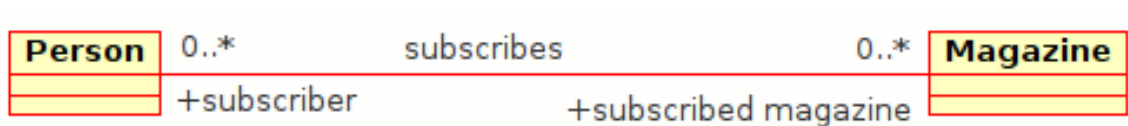
2 RELACIONAMENTOS

2.1 Entre instâncias

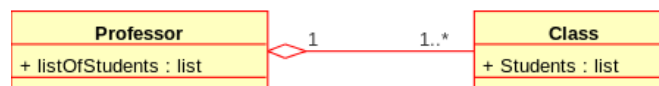
Os relacionamentos possuem **multiplicidade**, indicando a aridade da relação:

- 0 Sem instâncias
- 0..1 Zero ou uma
- 1 Exatamente uma
- 1..1 Exatamente uma
- 0..* Zero ou muitos
- * Zero ou mais
- 1..* Uma ou mais
- N..* N ou mais

Associações

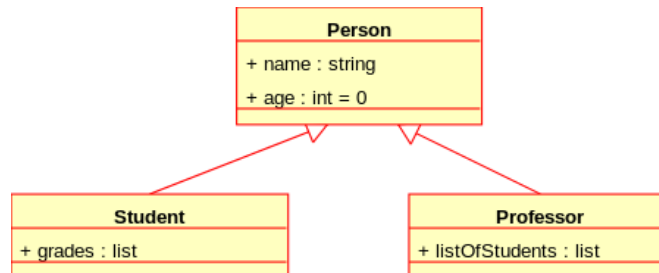


Composições

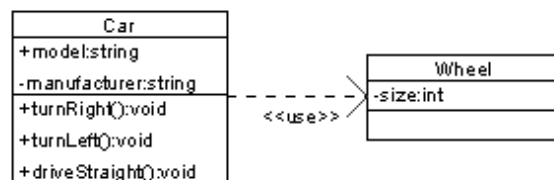


2.2 Entre classes

Herança/generalização



Dependência



De onde SAIU a especificação UML ? Como sei que atributos estão contidos em classes ?

A especificação completa de todos os diagramas pode ser encontrada no site da OMG. www.omg.org, A versão 2.5.1 possui 796 páginas. Nós iremos

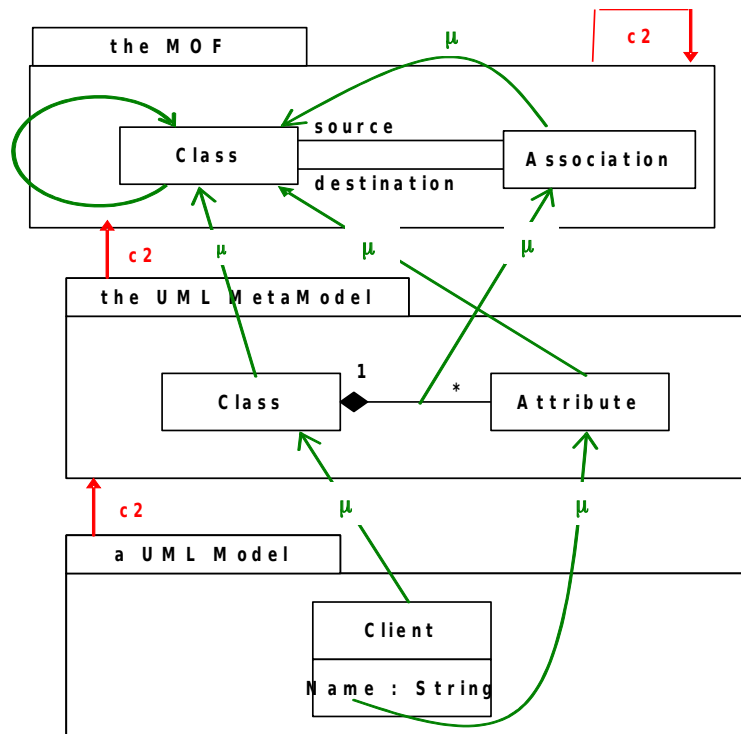
DEFINIÇÃO DE UMA ARQUITETURA DE PROJETO/MODELAGEM

Arquitetura em 3 níveis (MDE – Model Driven Engineering)

Modelo → contém os elementos que são especificados. O projeto do software.

Metamodelo → contém a definição dos elementos possíveis de um modelo UML.

Metametamodelo → contém a definição dos elementos possíveis do metamodelo UML.



Este esquema é similar na definição de notações e linguagens, em várias plataformas e formalismos.

XML document – schema – xschemaDEf

SQL – schema – data model

Java – Java grammar – EBNF

UML – UML metamodel – MOF

Um modelo qualquer – um metamodelo Ecore – metamodelo Ecore