Análise de Sistemas

Aula 5

Prof. Emerson Klisiewicz

Contextualização

Aula 5

- Análise Orientada a Objetos
- Introdução à UML – Histórico e Visão Geral
- Ferramentas CASE

O Sucesso

Clientes satisfeitos

- Eles estão satisfeitos quando você:
  - atende às expectativas
  - entrega no prazo
  - entrega tudo dentro do orçamento

- E para isso acontecer precisamos estar auxiliados por uma boa metodologia e ferramentas CASE

Instrumentalização

Histórico de Orientação a Objetos

- A OO surgiu no final da década de 60, quando dois cientistas dinamarqueses criaram a linguagem Simula (*Simulation Language*)

- 1967 – Linguagem de Programação Simula-67 – conceitos de classe e herança
- Início dos anos 90 ⇨ Paradigma de Orientação a Objetos
  - Abordagem poderosa e prática para o desenvolvimento de *software*

Análise Orientada a Objetos

- O modelo de casos de uso fornece uma perspectiva do sistema a partir de um ponto de vista externo
- De posse da visão de casos de uso, os desenvolvedores prosseguem com o sistema

- A funcionalidade externa de um sistema orientado a objetos é fornecida através de colaborações entre objetos

- Externamente, os atores visualizam resultados de cálculos, relatórios produzidos, confirmações de requisições realizadas etc.
- Internamente, os objetos colaboram uns com os outros para produzir os resultados

- O diagrama da UML utilizado para representar o aspecto **maior** da orientação a objetos é o diagrama de classes

Análise Orientada a Objetos – Conceitos

- Criou o conceito de objeto, que é um tipo de dado com uma estrutura e operações para manipular esta estrutura

- Classe: é um tipo definido pelo usuário que contém o molde, a especificação para os objetos
  - Todo objeto é uma instância de uma classe
  - Possui propriedades (**atributos**) e comportamento (**métodos**)

UML

- UML (*Unified Modeling Language*) – Linguagem de Modelagem Unificada
- É uma linguagem de modelagem (visual), não uma linguagem de programação

- Permite a utilização de diagramas padronizados para especificação e visualização de um sistema
- É uma linguagem de modelagem não proprietária

UML – Histórico

- Surgiu da união de três metodologias de modelagem:
  - Método de Booch, de Grady Booch

- Método OMT (*Object Modeling Technique*) de Ivar Jacobson
- Método OOSE (*Object Oriented Software Engineering*) de James Rumbaugh

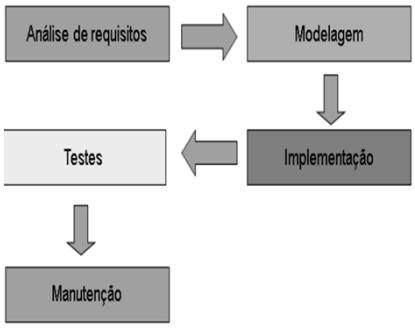
- A primeira versão foi lançada em 1996 e em 1997 a UML foi adotada pela OMG (*Object Management Group* – Grupo de Gerenciamento de Objetos) como padrão em modelagem

UML – Por quê?

- Bons modelos são essenciais para a comunicação entre os times de projetos e para assegurar a beleza arquitetural
- Facilita a programação

- Todo o time entende a modelagem, facilitando assim a manutenção
- Ter um rigoroso padrão de modelagem é fator essencial para o sucesso do projeto

UML – Onde?



UML – Modelagem

- Modelos proporcionam:
  - visualização do sistema
  - especificação da estrutura ou comportamento do sistema
  - guia para a construção do sistema
  - documentação das decisões tomadas

UML – Modelagem – Tipos

- Tipos de modelagens
  - Estrutural
  - Comportamental

UML – Diagramas

- Representação gráfica de um conjunto de elementos
- A UML, conforme a modelagem, possui alguns diagramas

- Estrutural (estática)
  - Diagrama de Classes
  - Diagramas de Objetos
  - Diagrama de Caso de Uso
  - Diagrama de Componentes

- Dinâmico (comportamental)
  - Diagrama de Estados
  - Diagrama de Atividades
  - Diagrama de Colaboração
  - Diagrama de Sequência

- Diagramas
  - Os documentos gerados em um processo de desenvolvimento são chamados de artefatos na UML
  - Os artefatos compõem as visões do sistema

- A UML define 15 diagramas
- Esta quantidade de diagramas é justificada pela necessidade de analisar o sistema por meio de diferentes perspectivas

- Cada diagrama fornece uma perspectiva parcial do sistema
- Ferramentas CASE auxiliam na construção e gerenciamento dos diagramas UML

Ferramentas CASE

- Ferramenta que oferece conjunto de serviços relacionados para apoiar uma ou mais atividades do processo de desenvolvimento de *software*

- Estudar ferramentas CASE é estudar:
  - como construir
    - ✓ definição de requisitos e arquitetura
  - como usar
    - ✓ processo de adoção, avaliação e seleção

Ferramentas CASE – Conceitos

- As ferramentas CASE podem ser:
  - horizontais: oferecem serviços utilizados durante todo o processo de *software*
  - verticais: utilizadas em fases específicas do processo de *software*

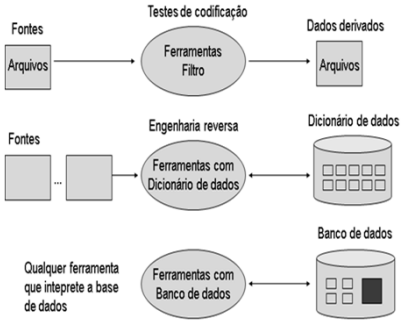
- Também podem ser classificadas de acordo com os serviços que oferecem, dentre as quais, cita-se:
  - Gerenciamento de configuração
  - Controle de qualidade
  - Programação
  - Documentação
  - Análise e projeto

Ferramentas CASE – Arquitetura

- A definição da arquitetura está intimamente relacionada ao contexto no qual a ferramenta atuará

- Uma ferramenta CASE deve ser flexível, com arquitetura modular para facilitar sua configuração para diferentes propósitos

Ferramentas CASE – Arquitetura – Exemplo



Ferramentas CASE – Exemplos

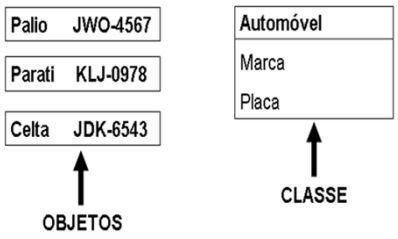
- Gerência de projetos
  - Microsoft Project
- Teste
  - Junit
  - Quality Center

- Ferramentas de métricas
  - USC-COCOMO
- Controle de versão
  - Git
  - Endevor

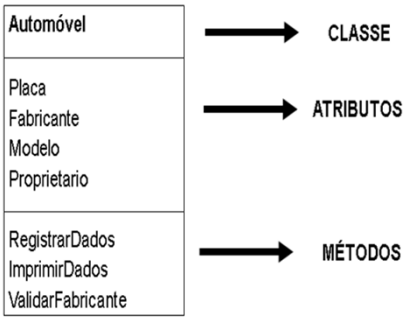
Aplicação

Análise Orientada a Objetos

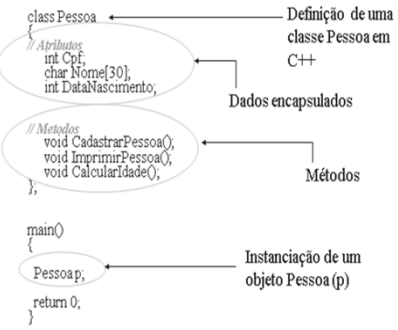
- Exemplo de classe e objetos



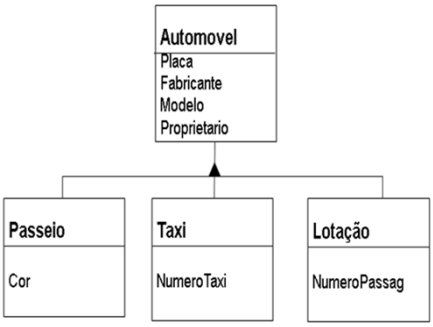
- Atributos e métodos



- Exemplo em C++



- Herança



UML – Diagramas

- Diagrama *Use Cases*
  - São especialmente importantes na organização e modelagem das principais funcionalidades de um sistema

- Diagrama de Classes
  - Os diagramas de classes são os principais diagramas estruturais da UML
  - Mostram classes, interfaces e seus relacionamentos

- Diagrama de Objetos
  - Representam instâncias estáticas de elementos dos diagramas de classes
  - São úteis para a modelagem de estruturas de dados complexas

- Diagrama de Sequência
  - Mostra um conjunto de objetos, seus relacionamentos e as mensagens que podem ser enviadas entre eles

- Diagrama de Colaboração
  - Mostra conjuntos de objetos, seus relacionamentos e as mensagens que enfatizam a organização dos objetos que trocam mensagens

- Diagrama de Estados
  - Mostra uma máquina contendo estados, transições, eventos e atividades
  - Estes diagramas são usados para modelar o comportamento de objetos (com comportamento complexo)

- Diagrama de Atividades
  - Destaca a lógica de realização de uma tarefa
  - Mostra o fluxo entre atividades

- Diagrama de Componentes
  - Mostra os componentes de *hardware* e *software* de uma aplicação e os relacionamentos entre eles
  - É usado para modelar o aspecto físico de um sistema

Ferramentas CASE

- O processo de adoção
  - Prover um nível apropriado de suporte tecnológico para os processos de desenvolvimento e manutenção de *software*

- Impactar positivamente sobre: produtividade, qualidade, padronização, documentação
- Induzir o uso geral e contínuo de ferramentas na organização e seus grupos

- Passos
  - Definição da necessidade
  - Avaliação e seleção de ferramentas
  - Condução de um esforço piloto
  - Tornar rotineiro o uso das ferramentas

Síntese

Pontos-chave

- Orientação a objetos apesar de antiga não era utilizada por falta de pessoas treinadas e ferramentas adequadas

- Mas hoje tal modelagem tornou-se uma abordagem poderosa e prática para o desenvolvimento de *software*
- A UML é uma linguagem de modelagem (visual) que permite a padronização de especificação e visualização de um sistema

- E temos as Ferramentas CASE, que apoiam a modelagem em todas as suas fases trazendo mais qualidade ao desenvolvimento de *software*