Análise e Projeto Orientados a Objeto

Objetivos

- Comparar e contrastar Análise e Projeto
- Definir Análise e Projeto Orientados a Objeto

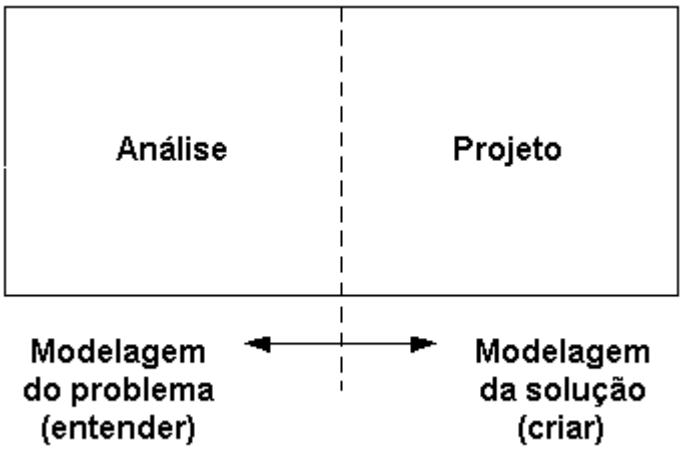
O que vamos fazer na disciplina?

- Saber uma linguagem de programação orientada a objeto (OO) não é suficiente para criar sistemas OO
  - Tem que saber Análise e Projeto OO (APOO)
  - Isto é, Análise e Projeto usando uma perspectiva de objetos
- Nesta disciplina, vamos nos concentrar na fase de Projeto, supondo que a Análise já foi feita
  - Na prática, a análise e o projeto são feitos em ciclos
  - Em cada ciclo, a análise vem *antes* do projeto
  - Ensinamos Projeto primeiro porque Análise requer mais maturidade e será feita em semestre posterior
- Usaremos a linguagem UML (*Unified Modeling Language*) para criar modelos (de análise e de projeto)
  - Um *modelo* é uma representação abstrata dos aspectos essenciais de um sistema
  - O que é "essencial" depende do momento da modelagem
  - A UML usa uma representação principalmente gráfica para representar os modelos
  - UML é *muito* popular hoje em dia para modelar sistemas
  - Não vamos ensinar UML nessa disciplina
    - Estude aqui (vou tentar achar um link melhor ...)
    - Resumo aqui
- Usaremos Design Patterns (padrões de projeto) para mostrar soluções abstratas para problemas que

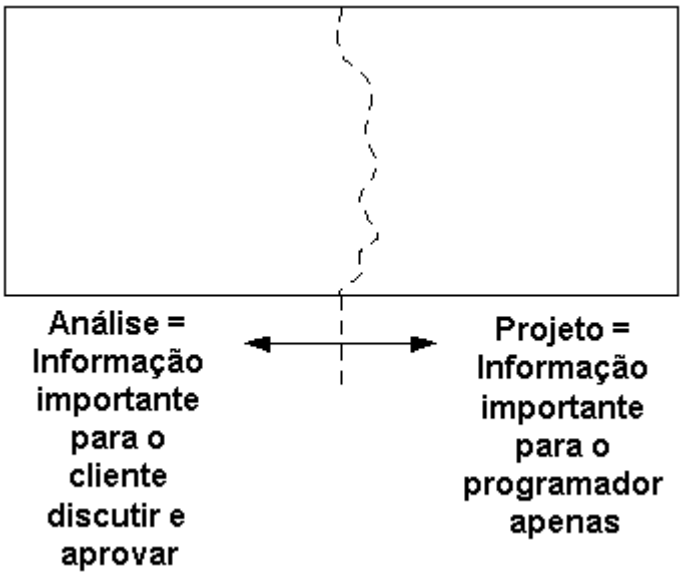
surgem freqüentemente durante o projeto de sistemas OO
    - Os patterns tratarão principalmente de:
        - Como atribuir responsabilidades a objetos (uma das atividades mais difícil no desenvolvimento de sistemas OO)
        - Como separar o que muda do que é constante numa determinada situação, com o objetivo de ganhar flexibilidade
            - Para evitar "o efeito gelatina"
- Análise e Projeto de Software são feitos usando-se um processo de desenvolvimento que mostra claramente quais são as etapas a seguir para produzir software de qualidade
    - Artefatos devem ser produzidos na várias fases e etapas do processo
    - Não falaremos muito de processos de desenvolvimento (tem outras disciplinas para isso)
    - Só são mencionados aqui para mostrar que Análise e Projeto não são atividades soltas mas fazem parte de uma "metodologia"

O que são Análise e Projeto?

- Diferenças entre análise e projeto: tem mais do que uma definição empregada
    - Primeira alternativa:
        - A análise modela o problema e consiste das atividades necessárias para entender o domínio do problema (*o que* deve ser feito). É uma atividade de *investigação*.
        - O projeto modela a solução e consiste das atividades de criação (*como* pode ser feito)



- Segunda alternativa:
  - A análise consiste de todas as atividades feitas com ou para o conhecimento do cliente. A informação produzida é aquela que o cliente deve discutir e aprovar
  - O projeto inclui as atividades que resultam em informação que interessa apenas ao programador
  - Com essa definição, a análise invade um pouco o "lado da solução", pois o cliente deve discutir alguns tipos de interações que ocorrerão na interface do usuário, etc.



- Observe portanto que não definição binária que isole "análise" de "projeto"
- Nesta disciplina, adotaremos a segunda alternativa, pois queremos associar as palavras "análise" e

"projeto" aos artefatos (*deliverables*) entregues nos final de cada fase

- Um modelo de análise deve ser aprovado pelo cliente e pode incluir alguma (pequena) discussão da solução, principalmente no que diz respeito à interface com usuário, etc.

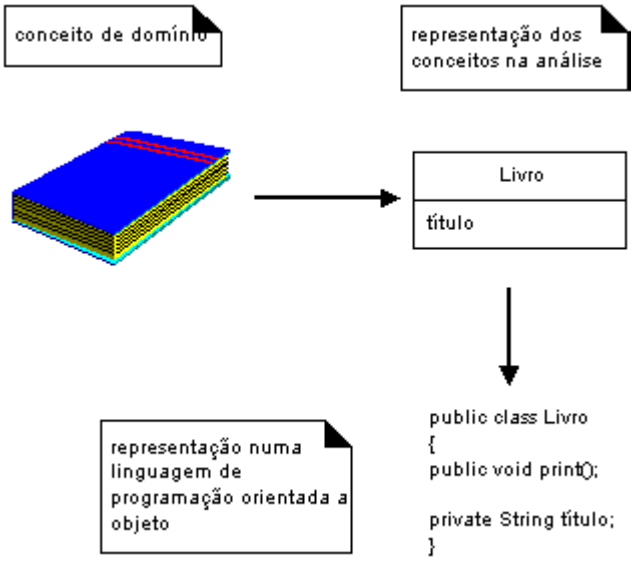
- Vamos mencionar também as fases de obtenção de requisitos, implementação e testes
  - A obtenção de requisitos é freqüentemente incluída na fase de análise ("análise de requisitos")

O que é Análise e Projeto Orientados a Objeto (APOO)?

- A perspectiva empregada é de *objetos* (coisas, conceitos ou entidades)
- Durante a Análise OO, a ênfase está em achar e descrever objetos (ou conceitos) no domínio do problema
  - Por exemplo, num sistema de informação para uma biblioteca, alguns dos conceitos são *Livro*, *Biblioteca*, *Usuário*.
  - Tais objetos podem ter atributos e responsabilidades
- Durante o projeto orientado a objeto, a ênfase está em achar objetos lógicos de software que poderão ser eventualmente implementados usando uma linguagem de programação OO
  - Tais objetos podem ter atributos e métodos
- Cuidado!
  - Não é verdade que haja correspondência 1-para-1 entre entidades no modelo de análise e entidades no modelo de projeto

- Pode haver entidade do modelo de análise que não será representado no Projeto (é raro)
- Pode haver entidade adicional no projeto (é freqüente)
  - Exemplo: Conexão de banco de dados, objeto controlador, cache de objetos, etc.

- Durante a construção (programação OO), os objetos do projeto são implementados e testados



intro programa