



Programação Orientada a Objetos

Prof. Hugo Marcondes

hugo.marcondes@ifsc.edu.br

Exemplo - Análise e Projeto OO

Baseado no Capítulo 1 do livro Applying UML and Patterns: An Introduction to
Object-Oriented Analysis and Design and Iterative Development, Third Edition,
Craig Larman, Prentice Hall, 2004

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina



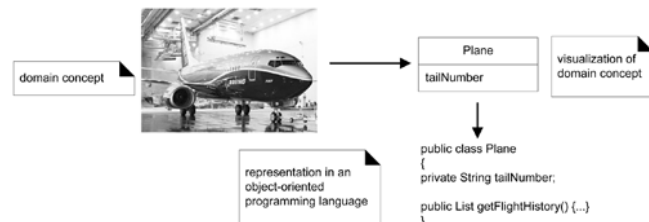
- O que é análise e projeto [analysis and design] ?
- Análise
 - Investigação do problema a ser tratado e seus requisitos
 - Não foca na solução do problema, e sim **ENTENDER** o problema
 - Ex: Um sistema de negociação on-line ?
 - Como ele será usado ?
 - Quais são as suas funções ?
 - Buscar os elementos do **domínio do problema**



- Projeto
 - Definição de uma **solução conceitual** que atenda aos requisitos do problema
 - **NÃO é a sua implementação**
 - Especificam a solução do problema, que poderá ser futuramente implementado.
- Ex: Esquema de banco de dados e objetos de software
 - Geralmente excluem detalhes de baixo nível ou óbvios (no contexto do quem irá utilizar o projeto).



- Análise orientada a objetos
 - Busca, localização e descrição dos objetos - ou conceitos - no **domínio do problema**
 - Ex: Sistema de Informações de voo :- Avião, Voo e Piloto.
- Projeto orientado a objetos
 - Definição de **objetos de software** e **como** eles **colaboram** para atender aos requisitos.
 - Ex: O objeto Avião, pode ter um atributo “tail number” e um método “getFlightHistory”



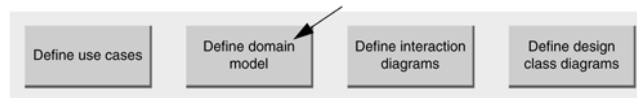




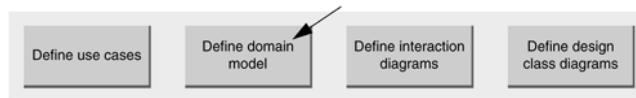
- Análise de requisitos
 - Estórias e cenários de como a aplicação é utilizada
 - Podem ser escritos como **casos de uso**

- **Caso de uso: Jogar o jogo de dados**

O jogador solicita rolar os dados. O sistema apresenta o resultado: Se o valor de face dos dados soma 7, o jogador ganha, caso contrário perde

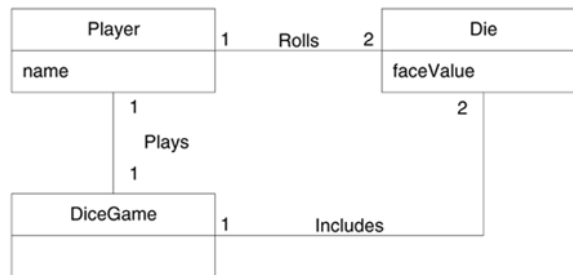


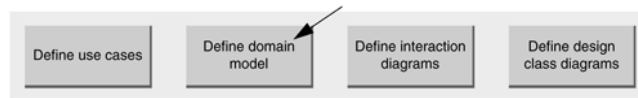
- A análise orientada a objetos preocupa-se em criar uma descrição do domínio a partir da perspectiva dos objetos.
- Identificação dos conceitos, atributos e associações que são considerados relevantes no domínio do problema
- Domain Model
 - Apresenta os conceitos ou objetos relevantes do domínio do problema



- **Caso de uso: Jogar o jogo de dados**

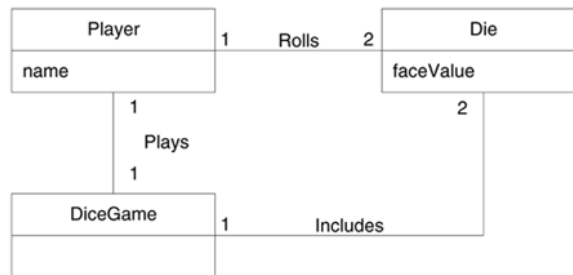
O **jogador** solicita rolar os **dados**. O **sistema** apresenta o resultado: Se o valor de face dos dados soma 7, o jogador ganha, caso contrário perde





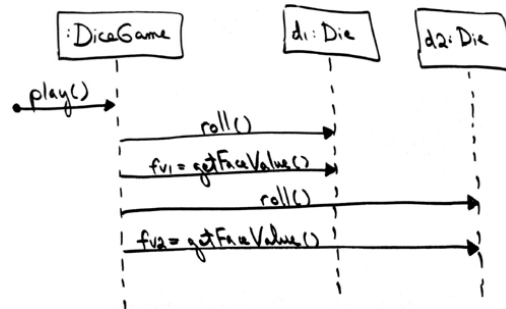
- **Caso de uso: Jogar o jogo de dados**

O jogador solicita rolar os dados. O sistema apresenta o resultado: Se o valor de face dos dados soma 7, o jogador ganha, caso contrário perde





- O design orientado a objetos se preocupa em definir objetos de software - suas responsabilidades e colaborações.
- diagrama de sequência
 - mostra o fluxo de mensagens entre objetos de software e, portanto, a invocação de métodos.





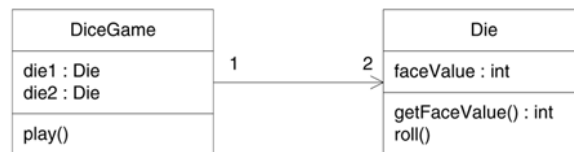
Define use cases

Define domain
model

Define interaction
diagrams

Define design
class diagrams

- Diagrama de Classes
 - Visualização estática das definições de classe
 - Ilustra os atributos e métodos das classes





Code monkey may refer to: A **term** for a computer programmer who isn't actually involved in any aspect of conceptual or design work, but simply writes **code** to specifications given.

- Wikipedia