

Análise Orientada a Objetos - Aula 02

Benay T. da L. de Carvalho - Universidade Norte do Paraná
RA: 3681235802

Diagramas de casos de Uso, classes e atividades.

- Diagrama de Casos de uso: é um diagrama comportamental com alta abstração, usado em requisitos.
- Diagrama de Classes: são os mais usados da UML, mostra a relação entre classes.
- Diagrama de Atividades: mostra o fluxo das ações e atividades de um sistema, é uma variação do caso de uso.

Diagrama de Casos de Uso

1. Elipse (caso de uso): verbo no infinitivo + substantivo
2. Ator (Objetos/Entidades): apenas o nome
3. Conexões: relações entre casos de uso
 - 3.1. Associação: associa diretamente um ator com caso de uso
 - 3.2. Generalização: herda elementos de um caso para outro(s)
 - 3.3. Extensão/Incluir: incluir é obrigatoriedade, extensão é opcional
4. Multiplicidade: quantidade de atores irão utilizar um caso de uso e vice-versa

Diagrama de Classes

1. Objeto: é a instância da Classe
2. Atributos: características do Objeto
3. Métodos: funcionalidades do Objeto
4. Encapsulamento
 - 4.1. Público(+) Public: todos têm acesso aos atributos
 - 4.2. Privado(-) Private: apenas a classe do objeto pode mexer nos atributos
 - 4.3. Protegido(#) Protected: funciona igualmente ao Privado com adicional que as classes filhas também têm acesso aos atributos
 - 4.4. Geralmente os atributos são privados e os métodos públicos mas isso pode variar

Tipos de Classe

Abstrata

- Representada pelo nome em itálico
- Não geram instâncias, servem apenas para representar hierarquia e generalização

Interface

1. Representada com «nomeInterface»
2. Representa um conjunto de elementos públicos que devem ser implementados por outras classes
3. Relacionamentos
 - Herança: herda funcionalidades da classe pai para classe filha. Representada por uma seta cheia
 - Implementação: as classes filhas irão implementar a classe interface pai, vão instanciar a classe pai, de certa forma. Representada por uma seta tracejada

Multiplicidade Multiplicidade representa os limites inferior e superior da quantidade de objetos aos quais outro objeto está associado. Esses valores podem variar de 0 à * (que representa muitos)

- 1 - Exatamente um
- 1...* - No mínimo 1, no máximo vários
- 0...1 - No mínimo 0, no máximo 1

Composição e Agregação

- Agregação: agregar um objeto a outro, de maneira não obrigatória. Similiar ao Extender do Caso de Uso. Representado por uma seta de losango.
- Composição: exige um objeto para outro existir, de maneira obrigatória. Similar ao Incluir do Caso de Uso. Representado por uma seta de losango cheia.

Estereótipos

1. «boundary»: representa a interface do sistema
2. «control»: controle da aplicação, serve de intermediário entre as classes boundary e entity
3. «entity»: contém informações geradas ou recebidas pelo sistema

Diagrama de Atividades Esses são os elementos mais comuns no Diagrama de Atividades:

1. **Inicial:** Representa o início do diagrama e é obrigatório(circulo fechado)
2. **Final:** Representa o fim de um fluxo, é usado para o finalizar o diagrama, pode ser usado mais de uma vez em diferentes circunstâncias (circulo fechado com contorno)
3. **Terminal:** Representa o fim de um fluxo, porém não encerra o diagrama, apenas aquele fluxo específico (circulo com X no meio)
4. **Decisão:** Elemento que indica uma condicional e algumas vezes causa um loop (losango)
5. **Transição:** Representa uma transição de uma atividade para outro elemento (seta)
6. **Fork/Join:** Fork divide uma transição em múltiplas transições. Join junta múltiplas transições em uma só, possuem efeitos opostos basicamente. (ambos são representados com uma linha dividindo o antes e depois das transições)
7. **Swimlanes:** Divisões colocadas no diagrama para organizar o momento das atividades.

Referências

1. Anotações da aula 2 de análise orientada a objetos
2. <https://pt.wikipedia.org/wiki/UML>
3. <https://spaceprogrammer.com/uml/>