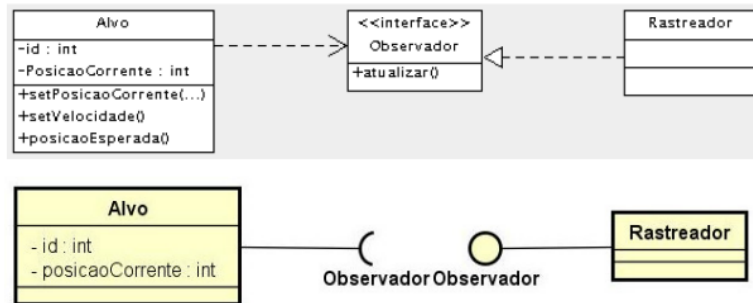


# Design de Software

## Diagramas de Classes Visão de Projeto – Interfaces, Pacotes e Visibilidade

- Interfaces

- Uma Interface é uma coleção de operações utilizadas para especificar um serviço de uma classe ou componente.
- As interfaces são empregadas para visualizar, especificar, construir e documentar a coesão interna do sistema.
- Declarando a interface, você pode estabelecer o comportamento desejado de uma abstração independente de qualquer implementação desta interface.
- Exemplo

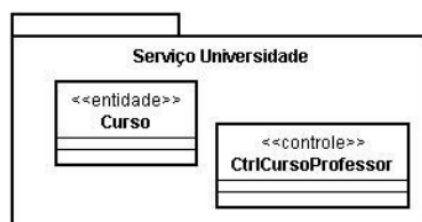


- A relação entre Alvo e Observador é uma relação de dependência, enquanto que a relação entre Observador e Rastreador é uma relação de realização.
- Uma relação de realização indica que a classe Rastreador implementa a interface Observador.

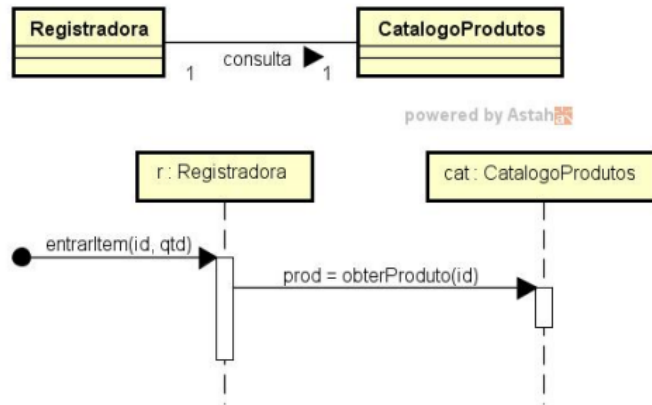
- Pacotes

- A maioria dos sistemas são compostos por muitas classes, neste caso então é necessário um mecanismo para agrupá-las, para obter uma melhor facilidade de uso, manutenção e reutilização.
- Neste caso, um pacote na visão lógica de um modelo é uma coleção de pacotes e/ou classes relacionados.
- Agrupando classes em pacotes, nós podemos ter uma visão de mais alto nível do modelo (i.e., os pacotes), ou podemos nos aprofundar no modelo, dando uma olhada no que está contido no pacote.
- Cada pacote contém uma interface que é implementada por um conjunto de classes públicas - aquelas classes com as quais os outros pacotes falam.
- O resto das classes em um pacote são classes de implementação - classes, que não se comunicam com as classes de outros pacotes.
- Na UML, pacotes são representados como pastas.
- Exemplo

- Seis classes: Curso, OfertaCurso, InformaçãoProfessor, OpçõesCursoProfessor, AdicionaOfertaCurso e ControleCursoProfessor.
- Elas caem em três grupos lógicos – coisas únicas para a universidade, coisas que contém informações sobre pessoas e coisas que são parte das interfaces com os atores. Nós podemos identificar os pacotes: Interfaces, ServicoUniversidade e InformacaoPessoa.

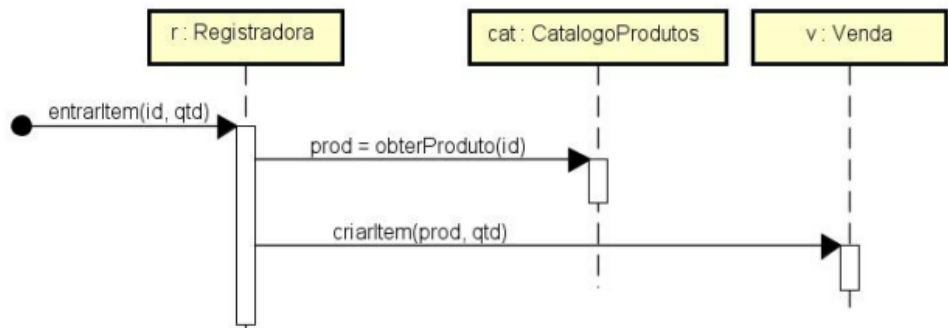






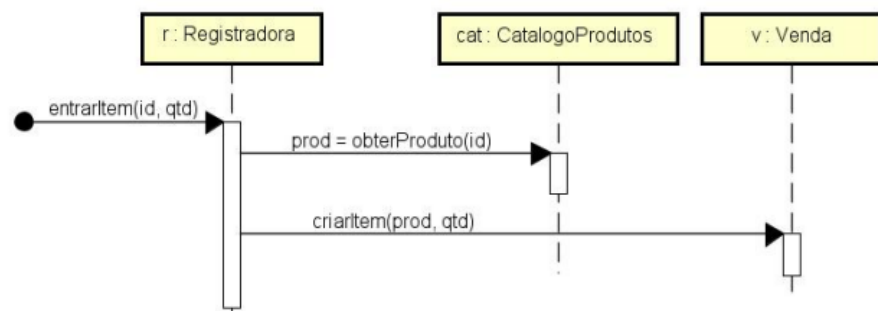
■  
○ Visibilidade por Parâmetro

- A visibilidade por parâmetro de A para B existe quando B é passado como parâmetro para um método de A
- No exemplo abaixo o objeto **prod** da classe **Produto** é visível por atributo para **v** da classe **Venda** pois é passado como parâmetro no método **criarItem()**



■  
○ Visibilidade Local

- A visibilidade local de A para B existe quando B é declarado um objeto local dentro de um método de A.
- No exemplo abaixo o objeto **prod** da classe **Produto** é visível local para **r** da classe **Registradora** pois é um objeto local no método **entrarItem()**



■  
○ Visibilidade Global

- A visibilidade global A para B existe quando B é declarado um objeto global para A.
- Para evitar objetos globais, pode-se usar o padrão Singleton (GOF)

○ Visibilidade e dependência

- As necessidades de visibilidade causam dependências entre as classes.
- Essas dependências são representadas nos diagramas de classes usando a relação de dependência