

Design de Software

Princípios de Orientação a Objetos

Baseado nos slides do Prof. Maurício Nacib Pontuschka (PUC-SP)

Design de Software

Princípios de Orientação a Objetos

O que é modelagem?

Um modelo é uma simplificação da realidade.



Objetivos da modelagem

- Ajudar a visualizar o sistema como gostaríamos que ele fosse
- Permitir a especificação da arquitetura e comportamento de um sistema
- Fornecer padrões de desenvolvimento que serve como guia de construção do sistema
- Documentar as decisões tomadas durante o processo de desenvolvimento

Por que modelar?

- Modelos de sistemas complexos são construídos porque nem sempre é possível compreendê-los em sua totalidade
- Modelos auxiliam o entendimento dos sistemas que construímos

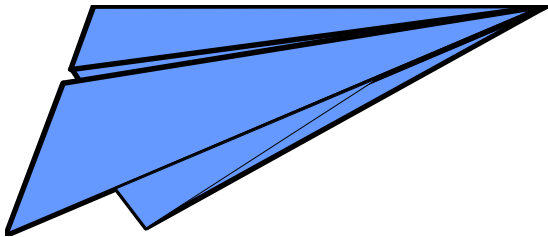
Design de Software

Princípios de Orientação a Objetos

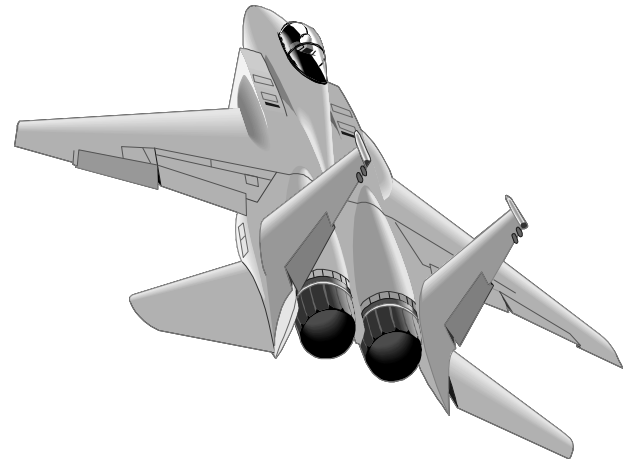
Importância da modelagem

Menos importante

Mais importante



Avião de papel



Jato militar

Design de Software

Princípios de Orientação a Objetos

Muitas equipes de desenvolvimento desenvolvem seus sistemas como se estivessem construindo aviões de papel

- Iniciam a codificação diretamente a partir das especificações do projeto.
- Trabalham durante horas e produzem mais código.
- Falta de um plano de arquitetura de software.
- Desenvolvimento fadado ao fracasso.

Design de Software

Princípios de Orientação a Objetos

Por que modelar?

- A modelagem é uma tarefa comum em projetos bem sucedidos.

Design de Software

Princípios de Orientação a Objetos

Model Driven Architecture (MDA)

- Uma abordagem de utilizar modelos no desenvolvimento de software.
 - Separar a especificação de uma operação dos detalhes da forma como o sistema utiliza os recursos da plataforma utilizada.
 - Especificar um sistema independentemente da plataforma que o suportará.
 - Especificar plataformas.
 - Escolher uma determinada plataforma para um sistema.
 - Transformar a especificação de um sistema específica para uma determinada plataforma.

Design de Software

Princípios de Orientação a Objetos

Pontos de visão do MDA

CIM – Computational Independent Model

O foco está no ambiente do sistema e seus requisitos.

PIM – Platform Independent Model

O foco está na operação do sistema, independentemente da plataforma.

PSM – Platform Specific Model

O foco está na utilização detalhada do sistema em uma plataforma específica.

Quatro princípios da modelagem visual

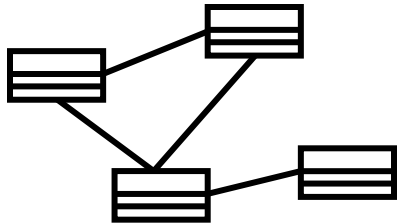
- O modelo que é criado influencia na forma como o problema é atacado.
- Todo modelo pode ser expresso em vários níveis de precisão.
- Os melhores modelos são os mais próximos à realidade.
- Nenhum modelo único é suficiente.

Design de Software

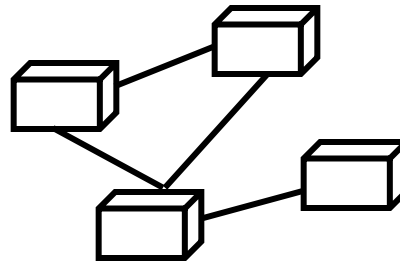
Princípios de Orientação a Objetos

Princípio 1: A escolha do modelo é importante

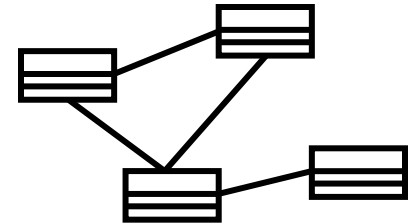
- Os modelos criados influenciam profundamente em como o problema é atacado e como a sua solução é elaborada.



Modelo de Processo



Modelo de Implantação



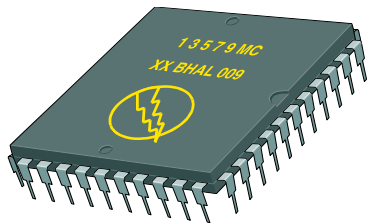
Modelo de Projeto

Design de Software

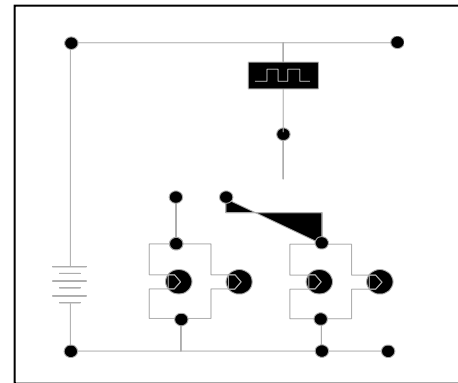
Princípios de Orientação a Objetos

Princípio 2: Níveis de precisão podem ser diferenciados

- Todo modelo pode ser expresso em diferentes níveis de precisão.
 - Quem visualiza o modelo e por que precisam visualizá-lo?



Visão dos
consumidores



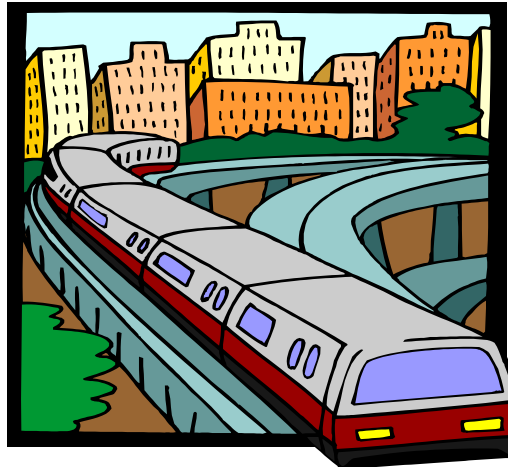
Visão dos
desenvolvedores

Design de Software

Princípios de Orientação a Objetos

Princípio 3: Os melhores modelos estão ligados à realidade

- Todos os modelos são simplificações da realidade.
- Um bom modelo reflete características reais.

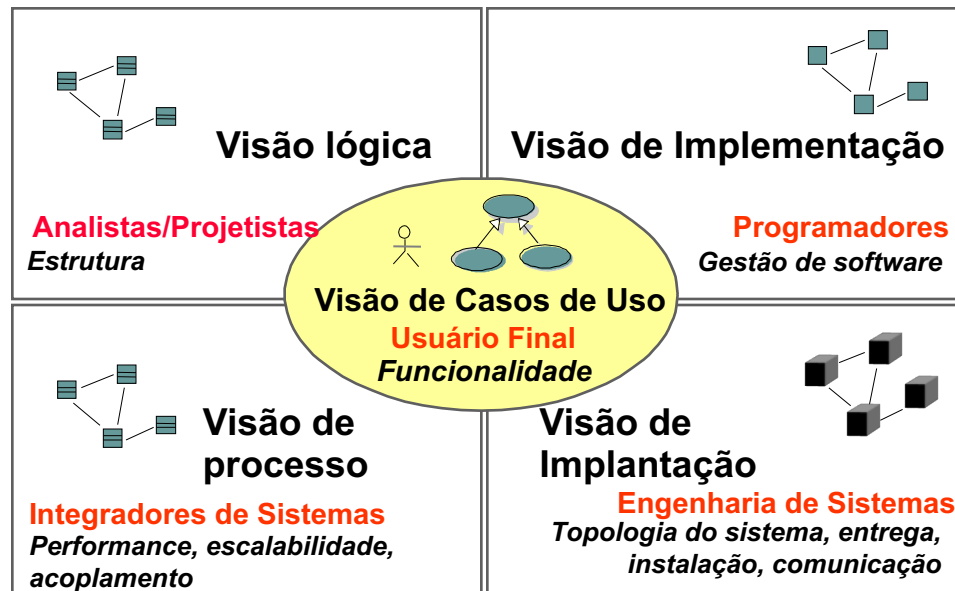


Design de Software

Princípios de Orientação a Objetos

Princípio 4: Nenhum modelo único é suficiente

- Todo sistema (não trivial) é melhor abordado por um conjunto de modelos.



Design de Software

Princípios de Orientação a Objetos

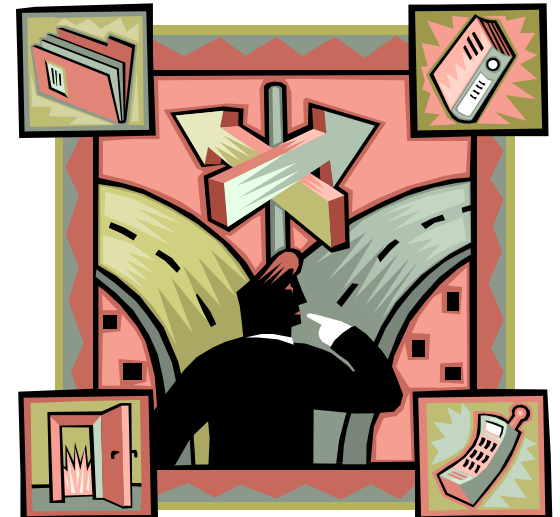
Próximos objetivos

- Descrever uma abstração, encapsulamento, modularidade e herança.
- Descrever a estrutura física de uma classe.
- Descrever o relacionamento entre uma classe e um objeto.
- Definir polimorfismo e generalização.

Design de Software

Princípios de Orientação a Objetos

- O que é um objeto?
- Quatro princípios da OO
- O que é uma classe?
- Polimorfismo e Generalização
- Organizando elementos de modelo

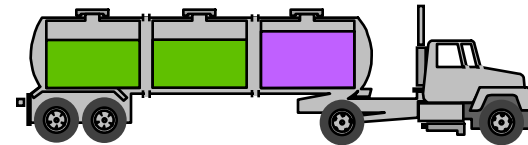


Design de Software

Princípios de Orientação a Objetos

Informalmente, um objeto representa uma entidade que pode ser física, conceitual ou de software.

Entidade física



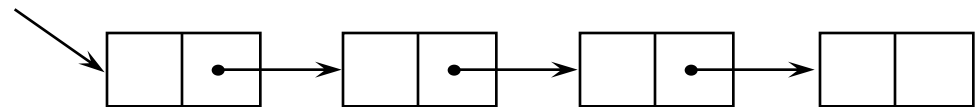
Caminhão

Entidade conceitual



Processo químico

Entidade de software



Lista ligada

Design de Software

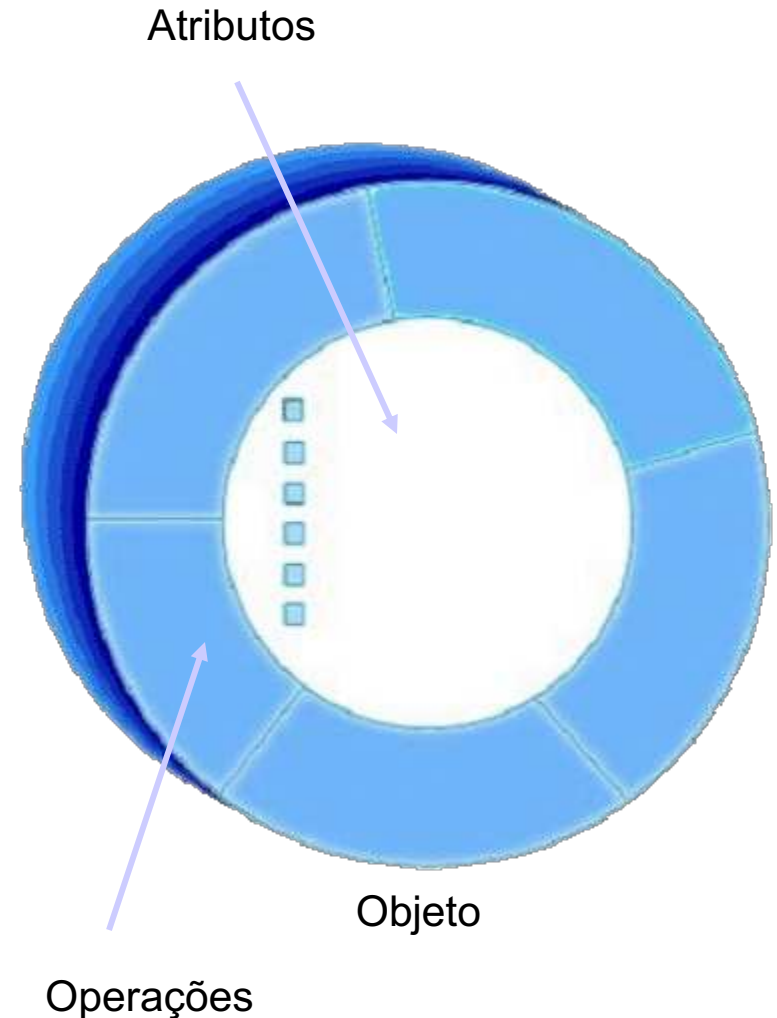
Princípios de Orientação a Objetos

Em uma definição mais formal:

Um objeto é uma entidade com uma fronteira bem definida e uma *identidade* que encapsula *estados* e *comportamento*.

Estado é representado por atributos e relacionamentos.

Comportamento é representado por operações, métodos e máquinas de estados.



Design de Software

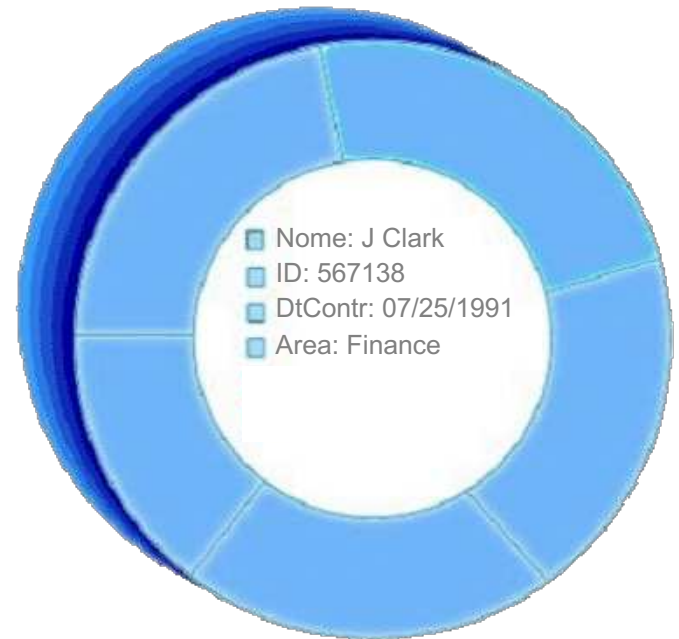
Princípios de Orientação a Objetos

Objetos possuem estados

O estado de um objeto é a condição ou situação durante o ciclo de vida de um objeto que satisfaz algumas condições, executa alguma atividade ou aguarda algum evento. O estado de um objeto normalmente muda ao longo do tempo.



Nome: J Clark
ID Funcionário: 567138
Contratação: 25 de Julho de 1991
Área: Finanças



Professor Clark

Design de Software

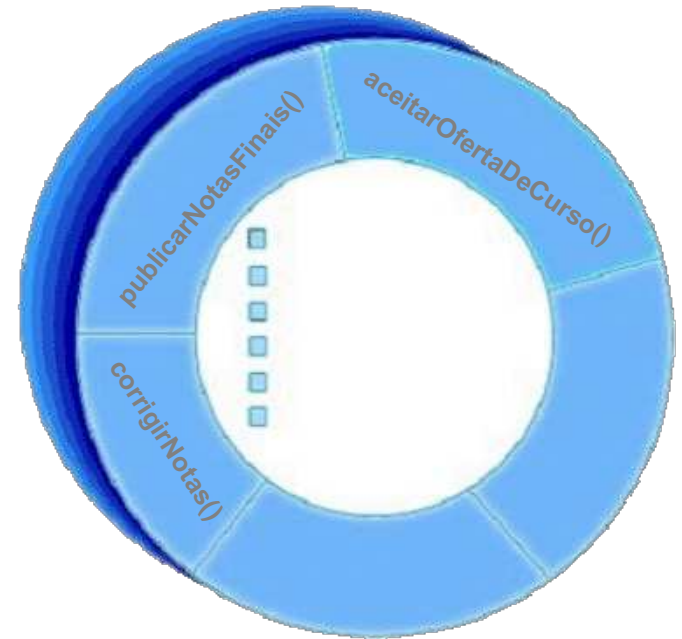
Princípios de Orientação a Objetos

Objetos possuem comportamento

O comportamento determina como um objeto age e reage. O comportamento observável de um objeto é modelado por um conjunto de mensagens que ele pode responder (operações que o objeto executa).



Comportamento do Professor Clark
Publicar notas finais
Aceitar oferta de curso
Corrigir notas



Professor Clark

Design de Software

Princípios de Orientação a Objetos

Um objeto possui identidade

Cada objeto possui uma identidade única, mesmo que o estado do objeto seja idêntico ao de outro objeto.



Nome: "J Clark"
Ensina: Biologia

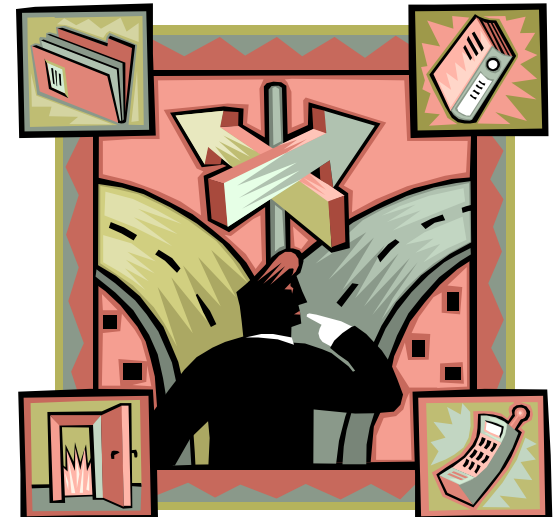


Nome: "J Clark"
Ensina: Biologia

Design de Software

Princípios de Orientação a Objetos

- O que é um objeto?
- Quatro princípios da OO
- O que é uma classe?
- Polimorfismo e Generalização
- Organizando elementos de modelo



Design de Software

Princípios de Orientação a Objetos

Princípios Básicos de Orientação a Objetos

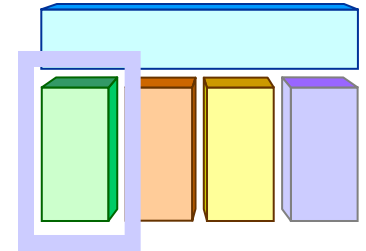


Design de Software

Princípios de Orientação a Objetos

O que é abstração?

- ◆ São características essenciais de uma entidade que a distingue de todos os outros tipos de entidade.
- ◆ Define uma fronteira relativa à perspectiva do observador.
- ◆ Não é uma manifestação concreta, denota a essência ideal de alguma coisa.



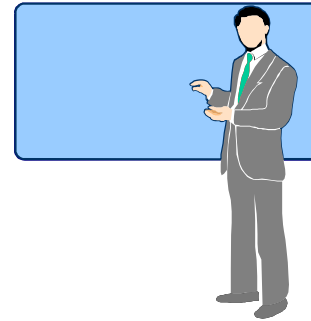
Design de Software

Princípios de Orientação a Objetos

Exemplos de Abstração



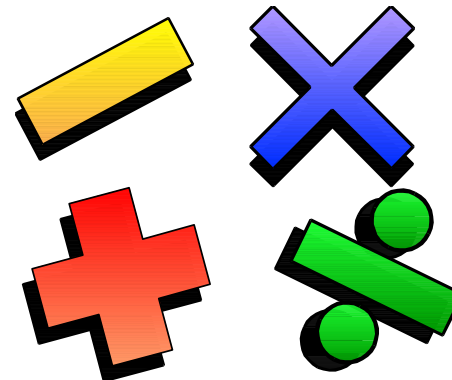
Estudante



Professor



Oferta de curso (9:00h,
segunda-quarta-sexta)



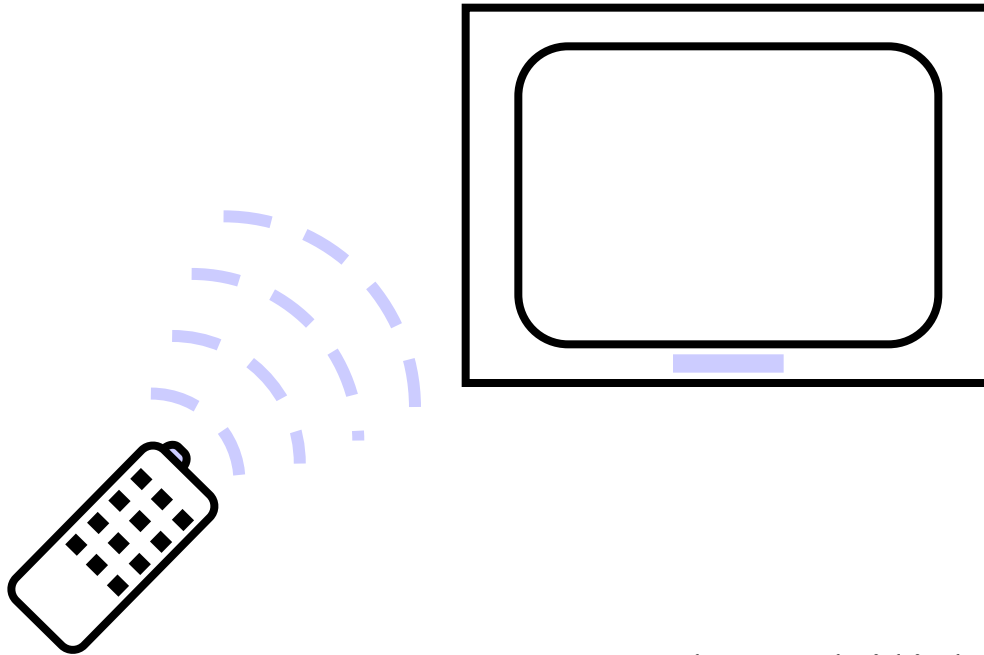
Curso (ex: Álgebra)

Design de Software

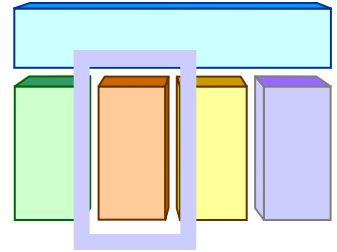
Princípios de Orientação a Objetos

O que é encapsulamento?

- ◆ Esconde a implementação dos clientes
- ◆ Clientes dependem da interface.



Favorece a adaptabilidade



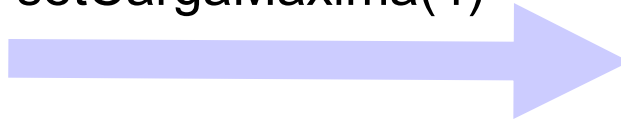
Design de Software

Princípios de Orientação a Objetos

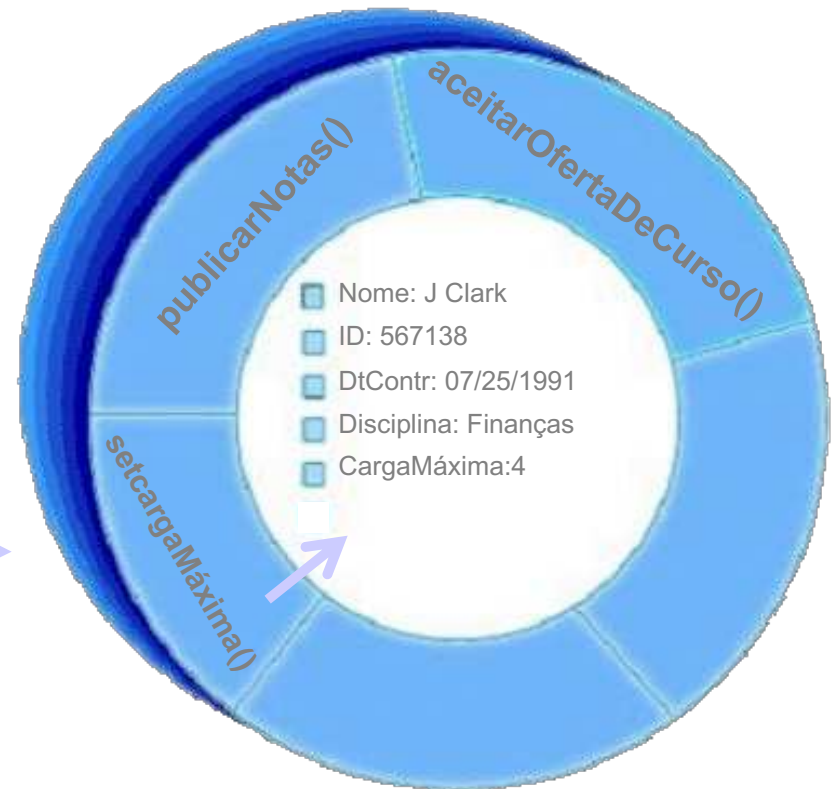
Exemplo de Encapsulamento

Professor Clark
precisa ser capaz de
ministrar quatro
turmas no próximo
semestre.

`setCargaMáxima(4)`



Professor Clark

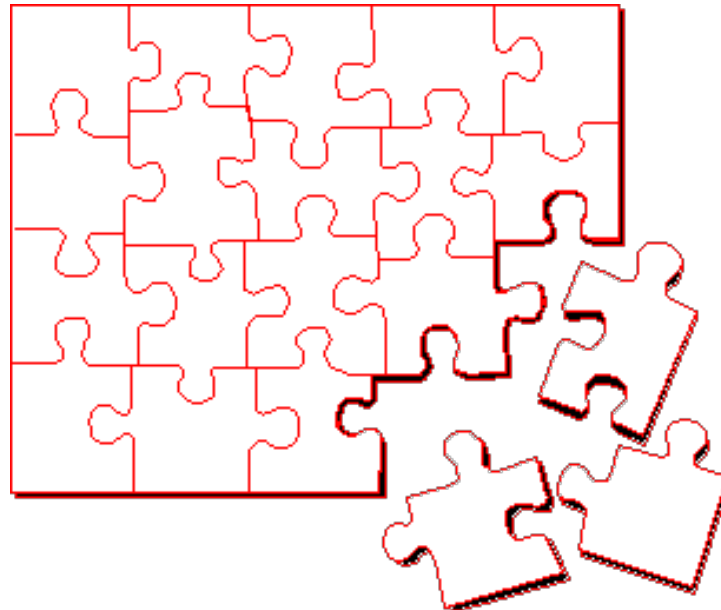
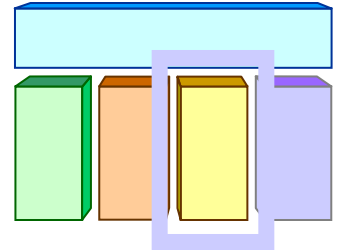


Design de Software

Princípios de Orientação a Objetos

O que é Modularidade?

- Quebra algo complexo em partes gerenciáveis.
- Auxilia as pessoas a entender sistemas complexos.

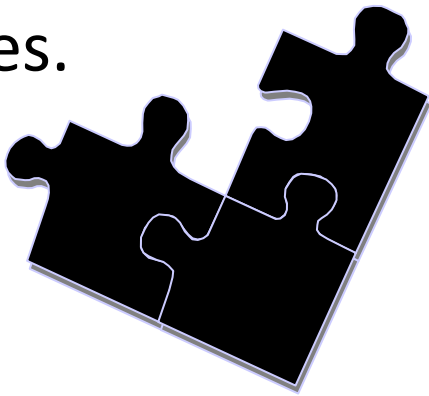


Design de Software

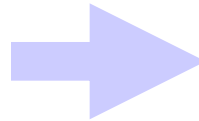
Princípios de Orientação a Objetos

Exemplo de Modularidade

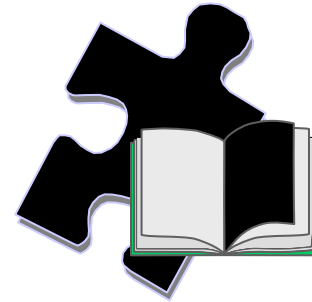
Por exemplo, quebrar um sistema complexo em módulos menores.



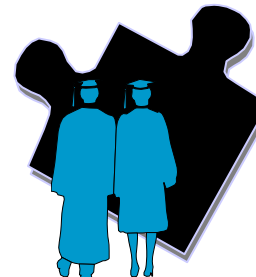
**Sistema de registro
de cursos**



**Sistema
de
cobrança**



**Sistema de
catálogo de
cursos**

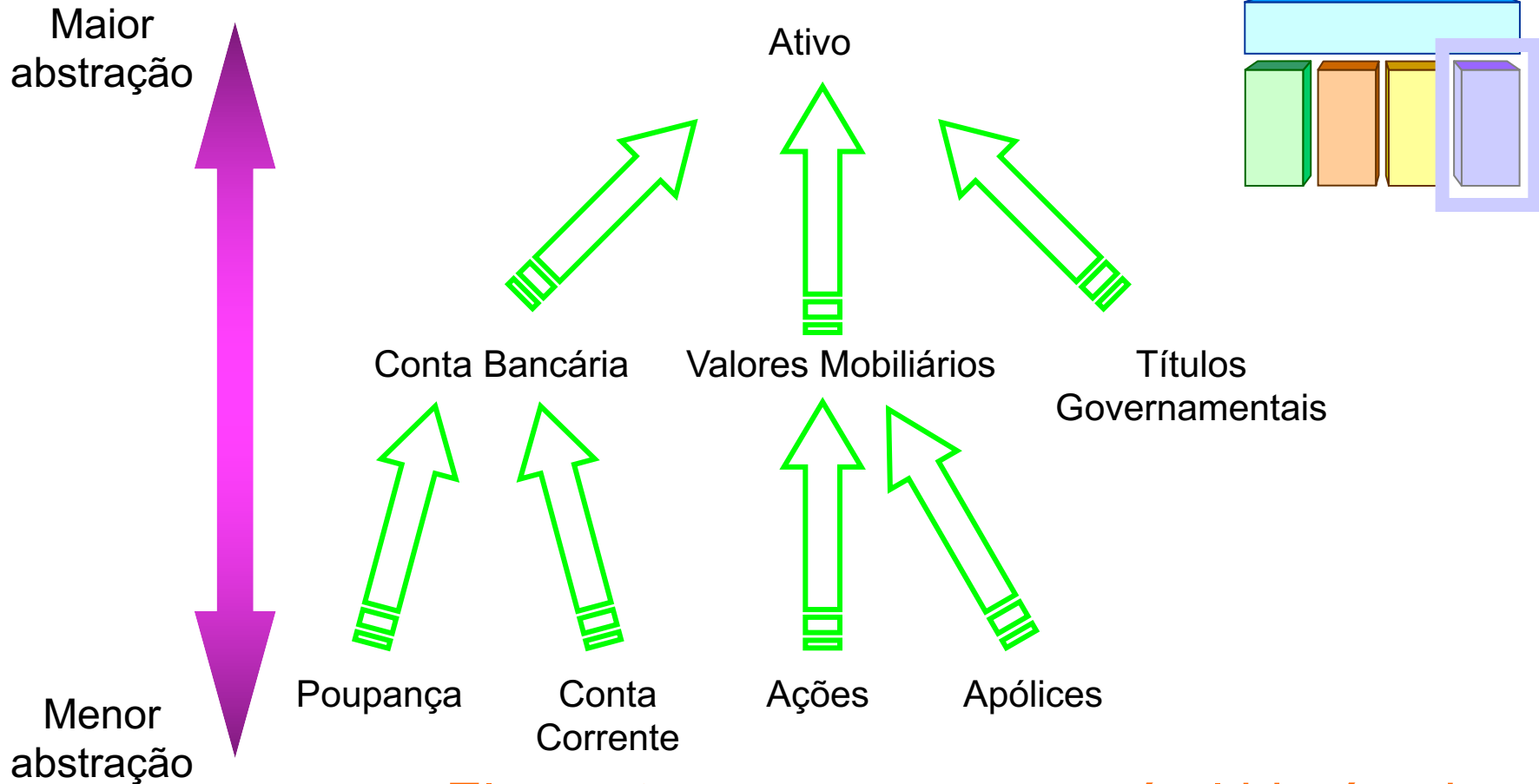


**Sistema de
gerenciamento
de estudantes**

Design de Software

Princípios de Orientação a Objetos

O que é hierarquia?



Elementos em um mesmo nível hierárquico devem estar no mesmo nível de abstração.

Design de Software

Princípios de Orientação a Objetos

Representação de Objetos em UML

Um objeto é representado por um retângulo com um nome sublinhado.



Professor J Clark

J Clark :
Professor

Objeto “nomeado”

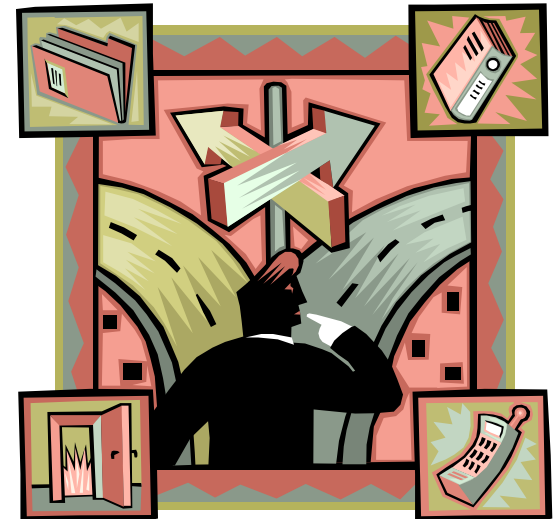
: Professor

Objeto “anônimo”

Design de Software

Princípios de Orientação a Objetos

- O que é um objeto?
- Quatro princípios da OO
- O que é uma classe?
- Polimorfismo e Generalização
- Organizando elementos de modelo



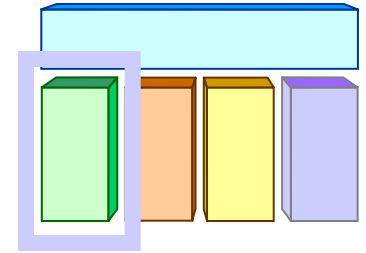
Design de Software

Princípios de Orientação a Objetos

O que é uma classe?

Uma classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

- Um objeto é uma instância de uma classe.
- Uma classe é uma abstração que:
evidencia características relevantes.
suprime outras características.



Design de Software

Princípios de Orientação a Objetos

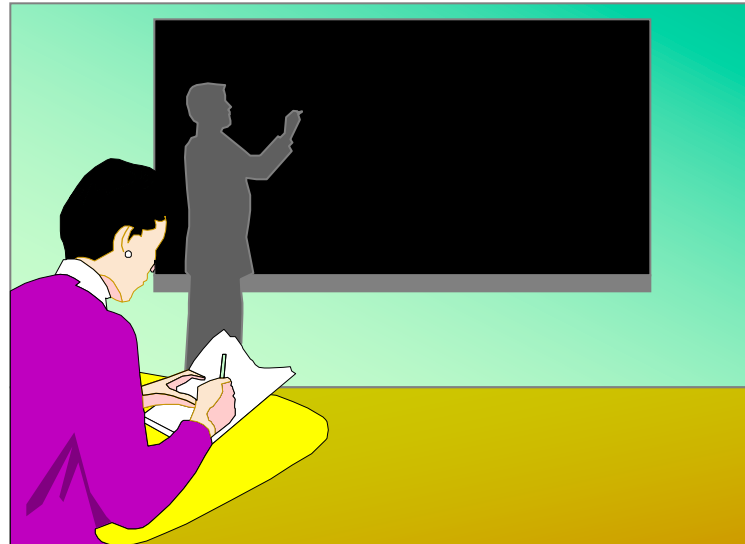
Exemplo de Classes

Classe

DISCIPLINA

Propriedades

Nome
Localização
Carga horária
Hora de início
Hora de término



Comportamento

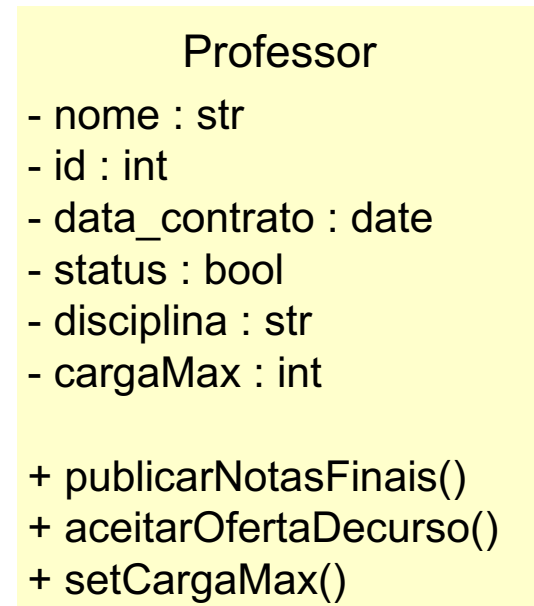
Adicionar estudante
Remover estudante
Obter lista de nomes
Indicar turma lotada

Design de Software

Princípios de Orientação a Objetos

Representando classes em UML

- Uma classe é representada utilizando um retângulo com três compartimentos:
 - O nome da classe
 - A estrutura (atributos)
 - O comportamento (operações)

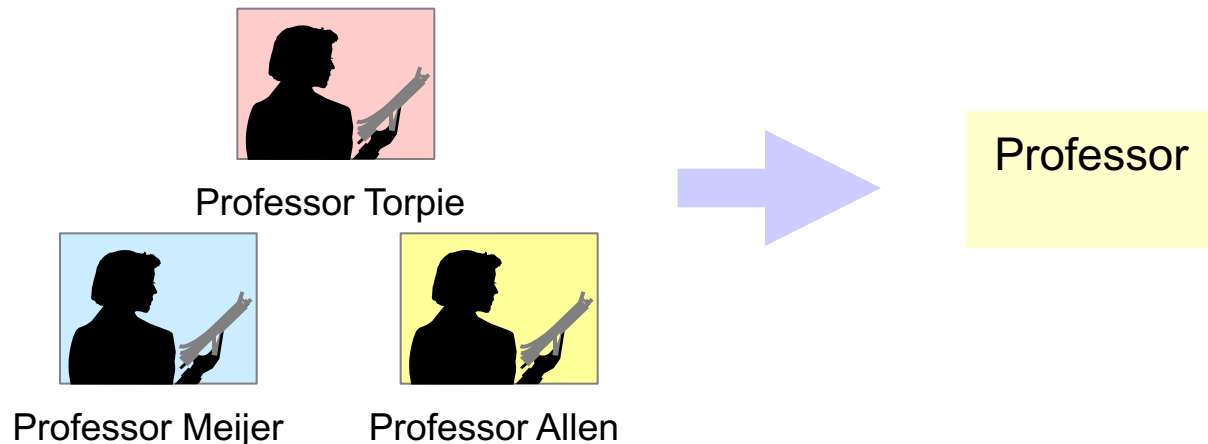


Design de Software

Princípios de Orientação a Objetos

Relacionamento entre classes e objetos

- Uma classe é uma definição abstrata de um objeto.
 - Ela define a estrutura e o comportamento de cada objeto da classe.
 - Se comporta como um molde (template) para criação de objetos.
- Classes não são coleções de objetos.



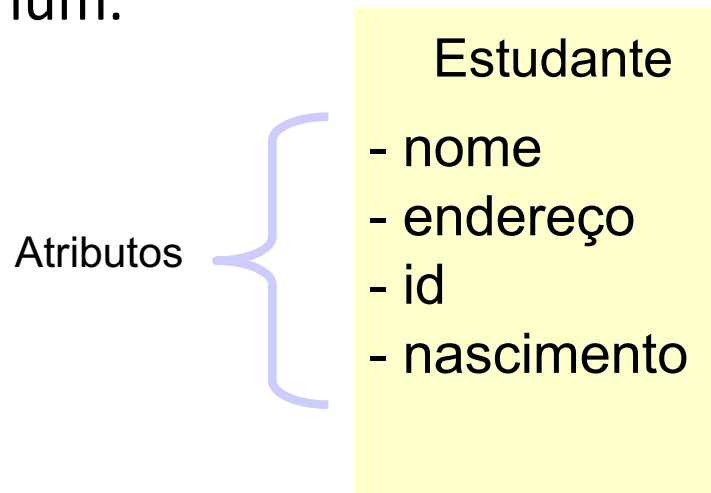
Design de Software

Princípios de Orientação a Objetos

O que é um atributo?

- Um atributo é uma propriedade nomeada de uma classe e descreve o valor que uma instância desta classe pode assumir.

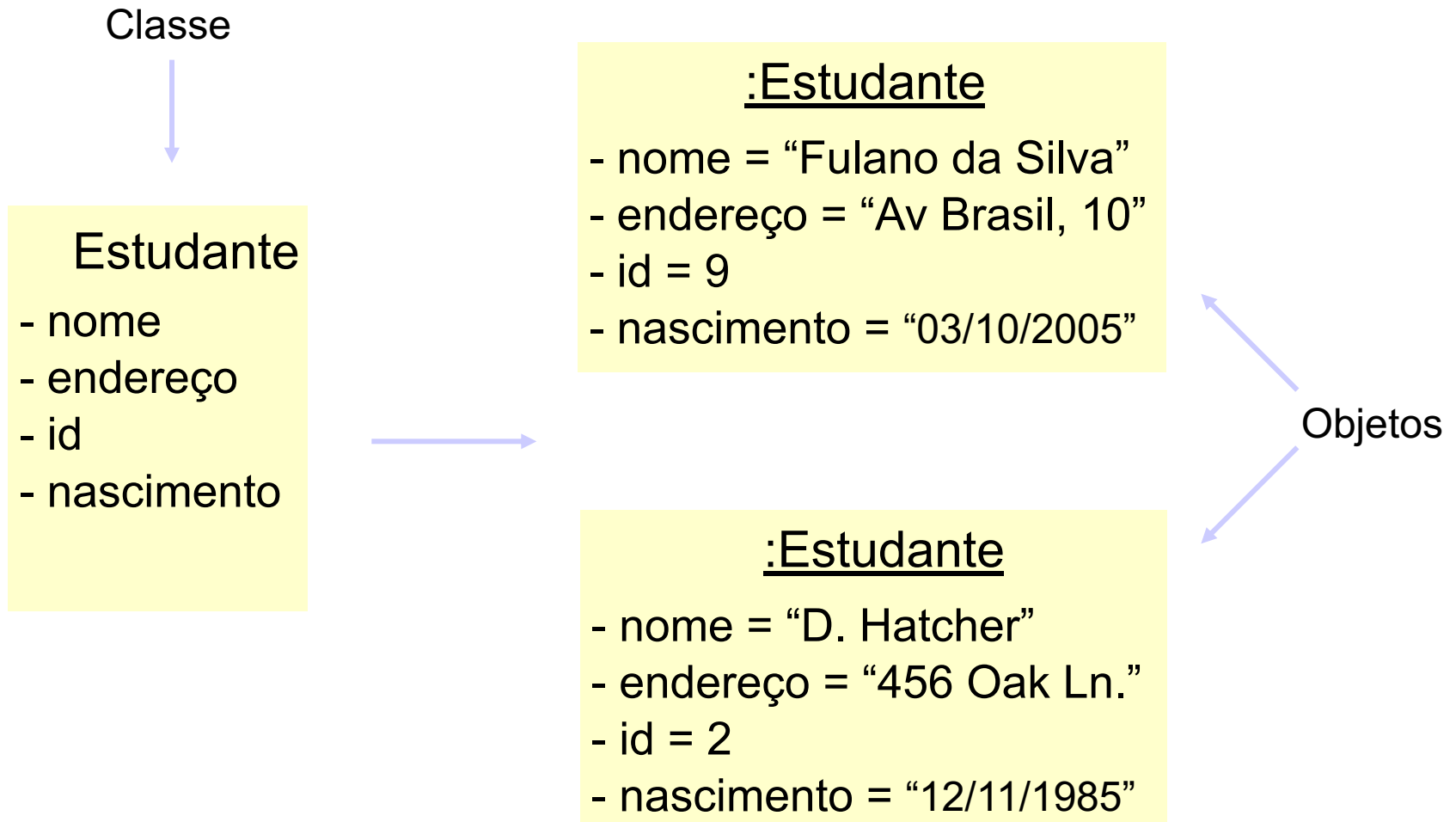
Uma classe pode possuir um número qualquer de atributos inclusive nenhum.



Design de Software

Princípios de Orientação a Objetos

Atributos em classes e objetos

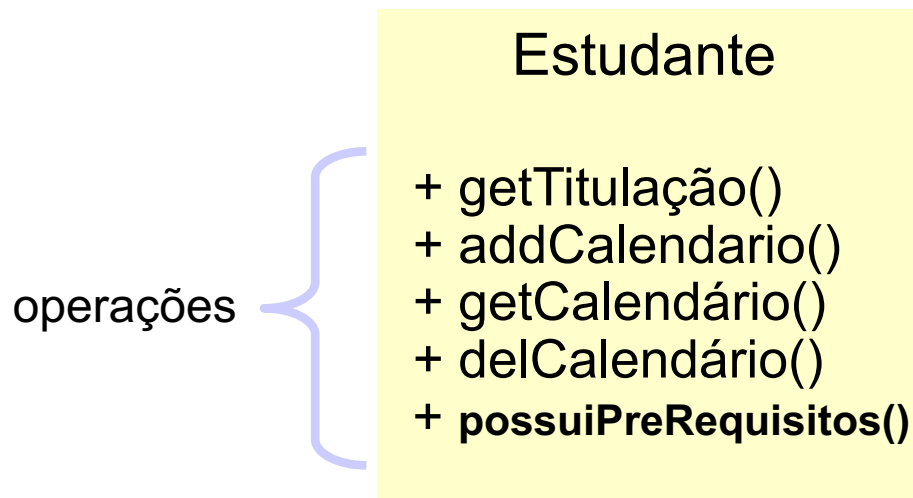


Design de Software

Princípios de Orientação a Objetos

O que é uma operação?

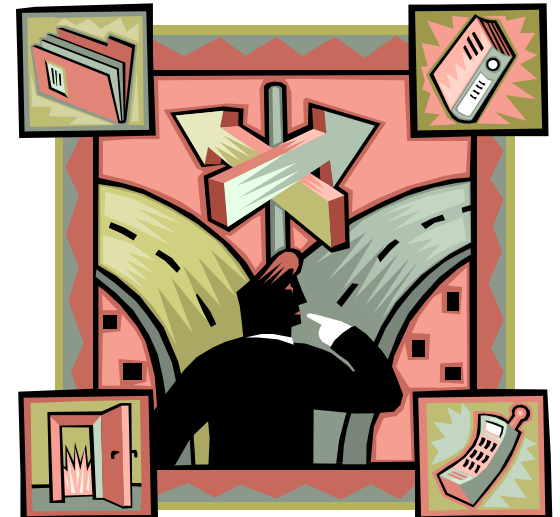
- Um serviço que pode ser requisitado de um objeto para afetar seu comportamento. Uma operação possui uma assinatura a qual define a forma e os parâmetros da operação.
- A classe pode ter nenhuma ou muitas operações.



Design de Software

Princípios de Orientação a Objetos

- O que é um objeto?
- Quatro princípios da OO
- O que é uma classe?
- Polimorfismo e Generalização
- Organizando elementos de modelo

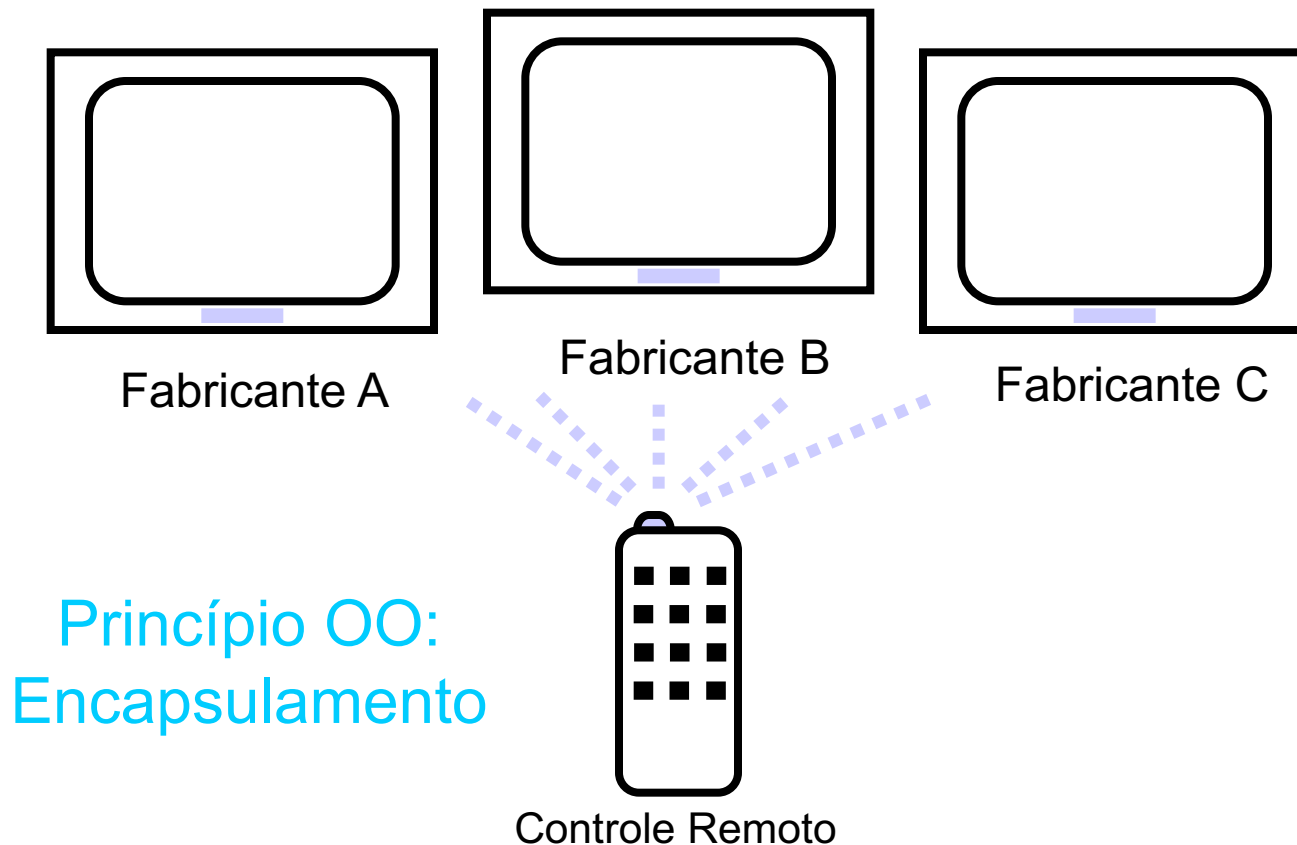


Design de Software

Princípios de Orientação a Objetos

O que é polimorfismo?

- ◆ Habilidade de esconder muitas implementações por trás de uma única interface.

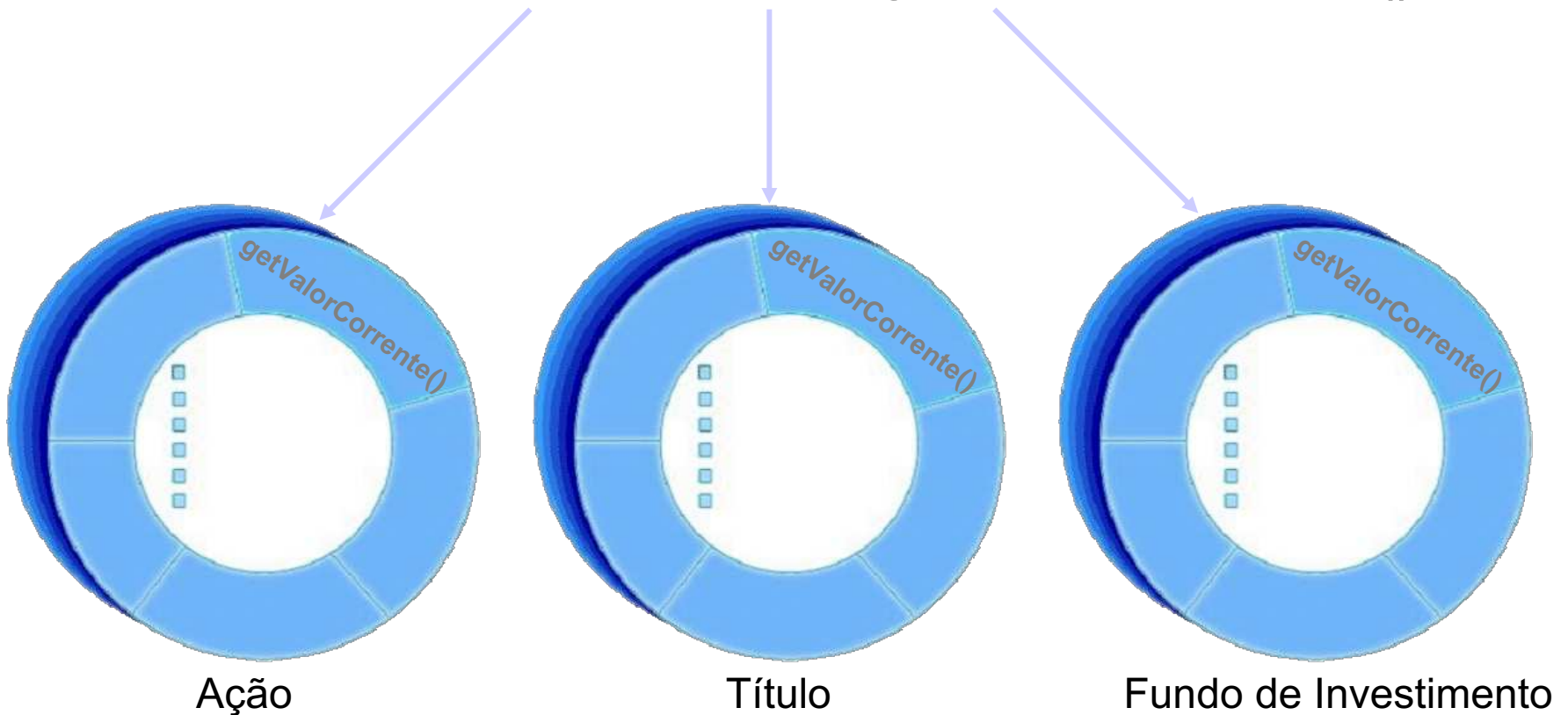


Design de Software

Princípios de Orientação a Objetos

Exemplo: Polimorfismo

`instrumentoFinanceiro.getValorCorrente()`



Design de Software

Princípios de Orientação a Objetos

O que é sobrecarga?

Uma mesma classe pode possuir operações com o mesmo nome e parâmetros diferentes.

Quando isso ocorre, dizemos que este método foi sobrecarregado.

Design de Software

Princípios de Orientação a Objetos

O que é sobreposição?

- Em uma hierarquia de classes, caso exista um método na sub-classe com a mesma assinatura de um método da classe pai, este é uma sobreposição ao método da super-classe.

Design de Software

Princípios de Orientação a Objetos

O que é Generalização?

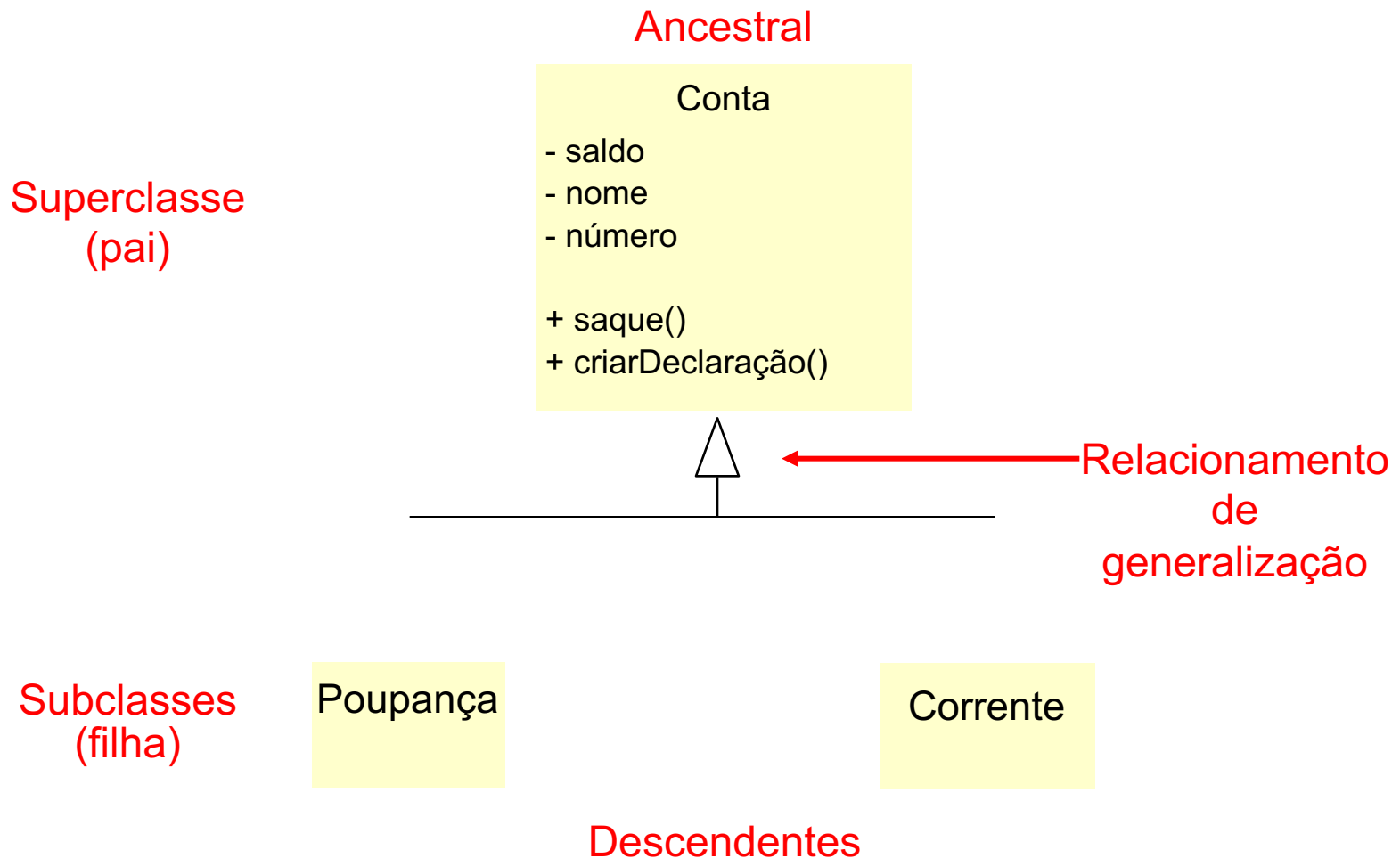
- Um relacionamento entre classes onde uma classe compartilha sua estrutura e/ou seu comportamento de uma ou mais classes.
- Define uma hierarquia de abstrações na qual uma subclasse herda elementos de uma hierarquia de superclasses, recebendo como herança suas características e comportamento.
 - Herança simples.
 - Herança múltipla.
- É um relacionamento “é um”.

Design de Software

Princípios de Orientação a Objetos

Exemplo: Herança Simples

Uma classe é herança de outra.

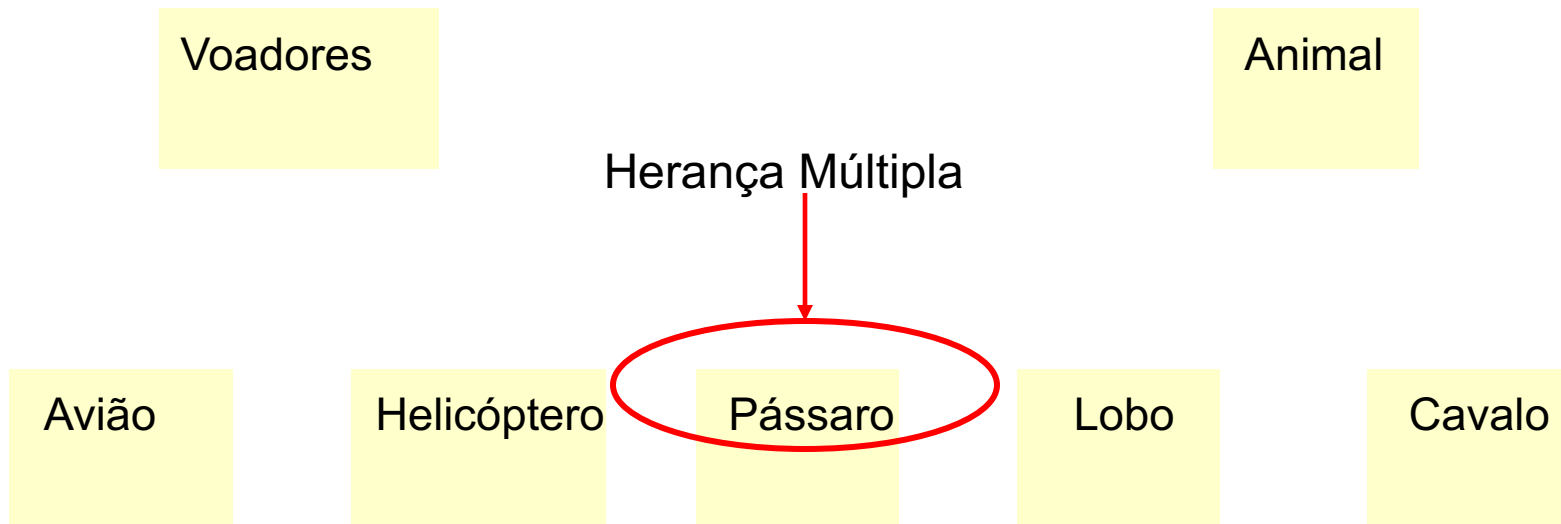


Design de Software

Princípios de Orientação a Objetos

Exemplo: Herança Múltipla

Uma classe pode ser herdeira de várias classes.



Use a herança múltipla somente quando realmente necessário e mesmo assim com muita atenção!

Design de Software

Princípios de Orientação a Objetos

O que é Herança?

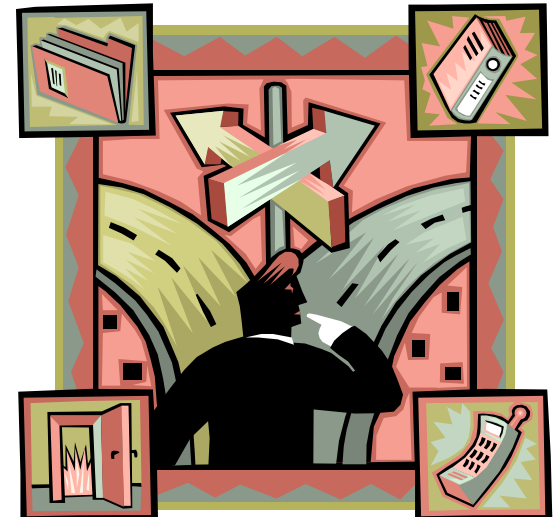
- Um subclasse herda os atributos, operações e relacionamentos da superclasse.
- Uma subclasse pode:
 - Adicionar atributos, operações e relacionamentos novos.
 - Redefinir operações herdadas. (Use com cuidado!)
- Atributos, operações e relacionamentos, são mostrados no nível mais alto aplicável da hierarquia.

A herança alavanca as similaridades através das classes.

Design de Software

Princípios de Orientação a Objetos

- O que é um objeto?
- Quatro princípios da OO
- O que é uma classe?
- Polimorfismo e Generalização
- Organizando elementos de modelo



Design de Software

Princípios de Orientação a Objetos

Artefato de Software

- Resultado de um conjunto de atividades proveniente das tarefas envolvidas no processo de desenvolvimento de um sistema de software.
- Isto é:
 - Um subproduto concreto de um produto de software;
 - Provê informações sobre e internamente ao software;
 - Um documento;
 - Um modelo;
 - Um executável.

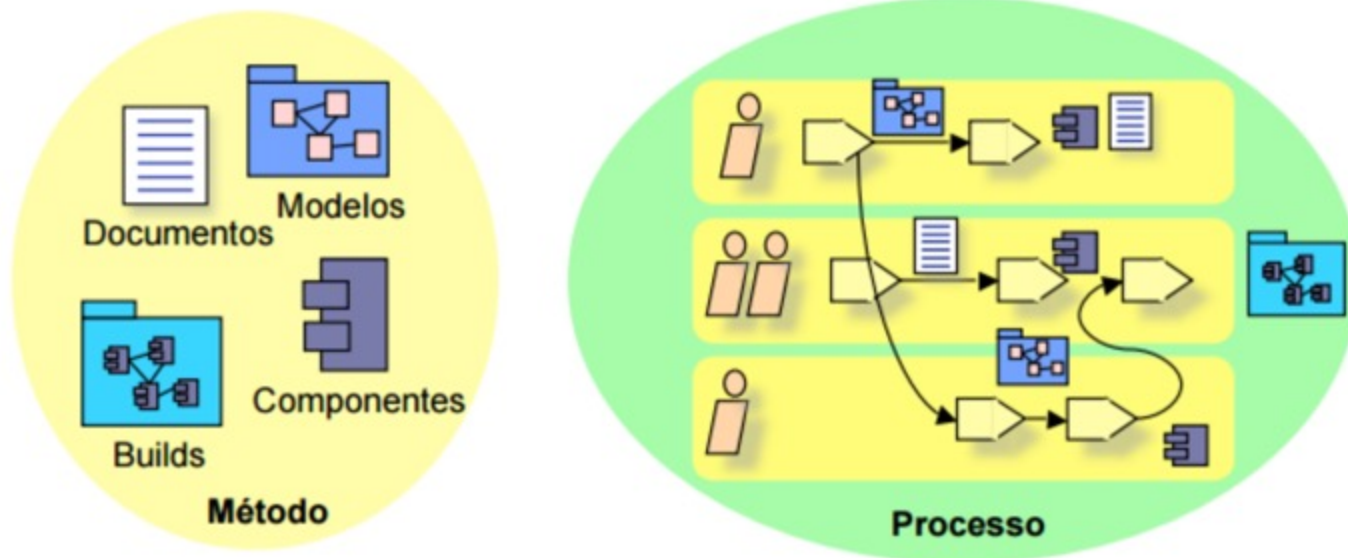
Design de Software

Princípios de Orientação a Objetos

Artefato de Software

- Exemplos:

- Documento de Requisitos
- Diagrama de Casos de Uso
- Diagrama de Classes



Design de Software

Princípios de Orientação a Objetos

O que é Pacote?

- Um mecanismo de proposta geral para organizar elementos em grupos.
- Um elemento de modelagem que pode conter outros elementos de modelagem.
- Um pacote pode ser usado:
 - Para organizar o modelo em desenvolvimento.
 - Como uma unidade de gestão de configuração.

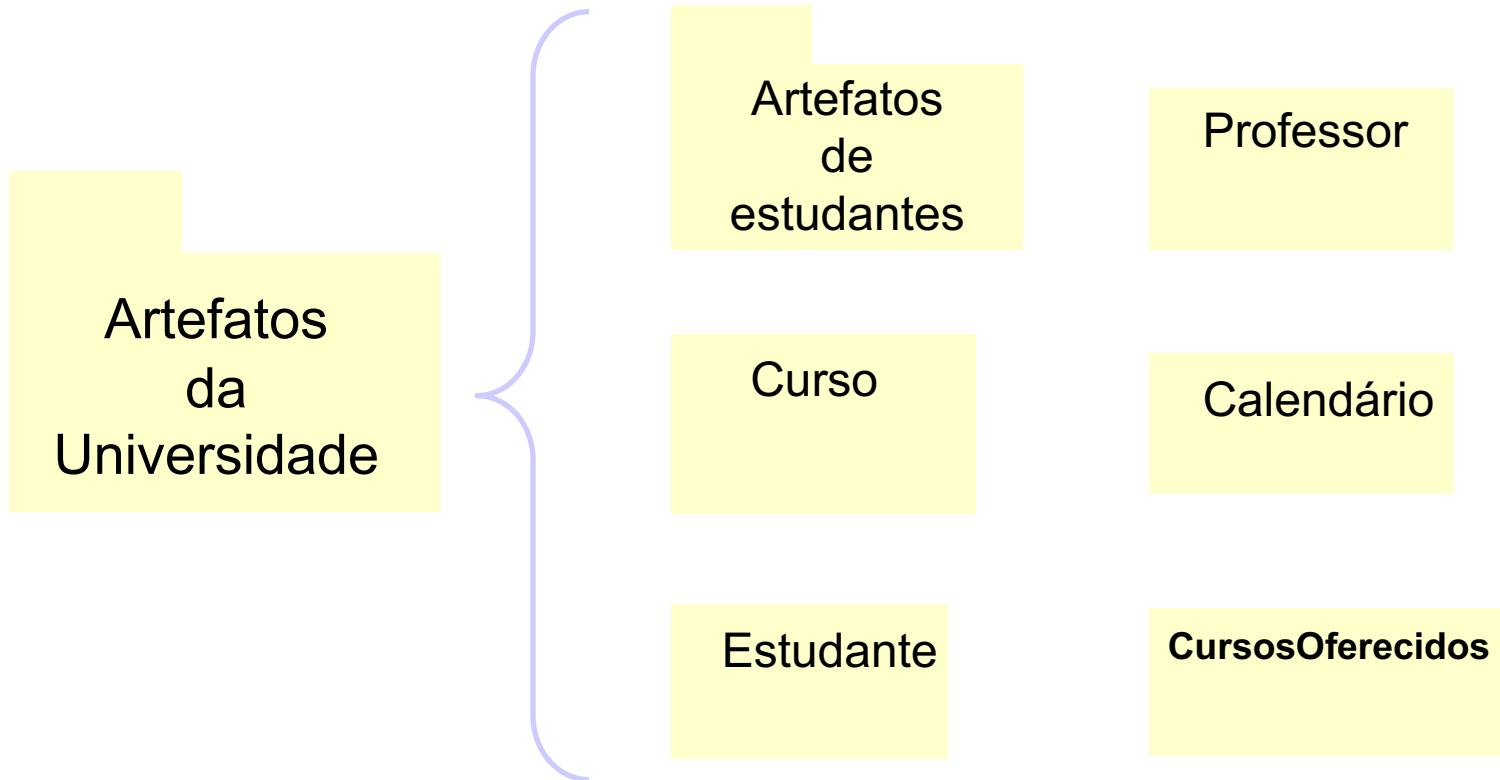


Design de Software

Princípios de Orientação a Objetos

Um pacote pode conter classes

O pacote Artefatos da Universidade contém um pacote e cinco classes.



Design de Software

Princípios de Orientação a Objetos

Revisão



- O que é um objeto?
- Quais são os quatro princípios de orientação a objetos? Descrever cada um.
- O que é uma classe? Como relacionar classes e objetos?
- O que é um atributo? Uma operação?
- Defina polimorfismo. Forneça um exemplo de polimorfismo.
- O que é generalização?
- Por que usar pacotes?