

Programação Orientada a Objetos

Turma 3

Carlos Henrique Bughi, MSc





Onde estamos?
(e para onde vamos)

- UML
- Mais UML
- Avaliação M1



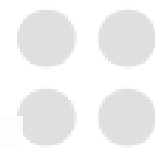


Aula 5

UML - Unified Modeling Language



Modelagem de Sistemas



.Deseja-se modelar e implementar uma aplicação para uma Loja Virtual. Para realizar esta tarefa você sabe que: a Loja Virtual deverá possuir catálogo de produtos e coleção de clientes

.Cada produto possui código numérico, nome e preço

.Cada cliente possui código numérico, nome, endereço e CPF

.A loja deverá ser capaz de listar todos produtos do catálogo, bem como todos clientes cadastrados

.Um cliente, ao fazer uma compra, relaciona uma lista de produtos e suas respectivas quantidades

.Ao final da compra, a loja apresenta pedido completo do cliente, incluindo seu nome, lista de produtos e suas quantidades, preços unitários, e total da compra





Modelagem de sistemas

.Para realizar a modelagem pense

.Quais são os objetos?

–Identificados através de substantivos

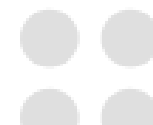
.Quais são os atributos dos objetos?

–Propriedades que o objeto deve ter

.Quais são os comportamentos (métodos) dos objetos?

–Identificados através de verbos



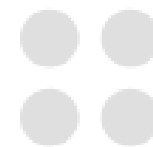


.Observações:

- .Ao identificar objetos, cuide com nomes utilizados para as classes
 - .Substantivos no singular: *Mensagem, Aluno, Produto*
 - .Algumas vezes associados a um adjetivo ou particípio: *AlunoEspecial, BufferDeEntrada, IconeAnimado*
 - .Evite a palavra “objeto”: *MailBoxObject (X)*
 - .Evite nomes genéricos: *Agente, Tarefa, Item (X)*
 - .Evite nomes que são verbos: *Imprimir, Calcular (X)*



Modelagem de sistemas



.Deseja-se modelar e implementar uma **aplicação** para uma **Loja Virtual**. Para realizar esta tarefa você sabe que: a Loja Virtual deverá possuir **catálogo de produtos** e **coleção de clientes**

.Cada **produto** possui código numérico, nome e preço

.Cada **cliente** possui código numérico, nome, endereço e CPF

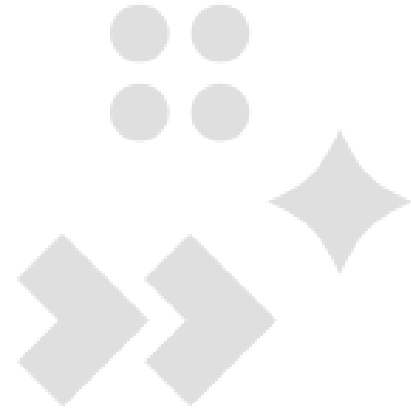
.A loja deverá ser capaz de listar todos produtos do catálogo, bem como todos clientes cadastrados

.Um cliente, ao fazer uma **compra**, relaciona uma lista de produtos e suas respectivas quantidades

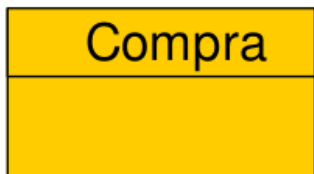
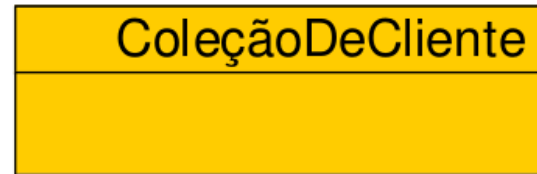
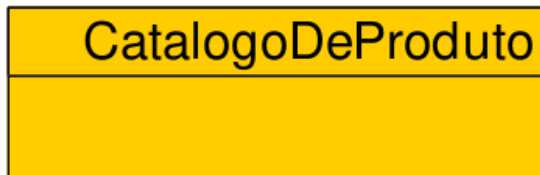
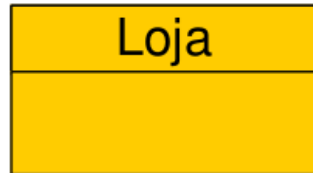
.Ao final da compra, a loja apresenta pedido completo do cliente, incluindo seu nome, lista de produtos e suas quantidades, preços unitários, e total da compra



Modelagem de sistemas



.Quais são os objetos? Definindo as classes

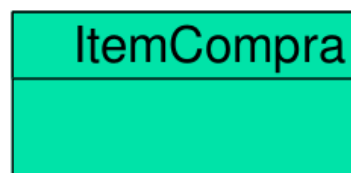
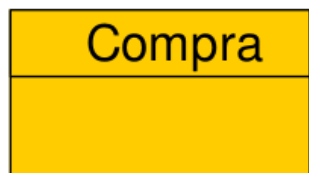
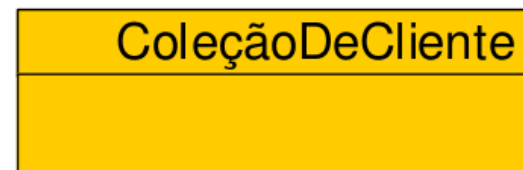
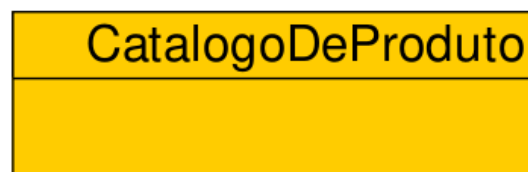
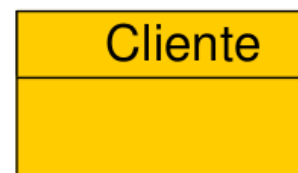
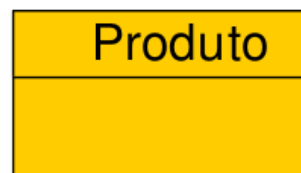
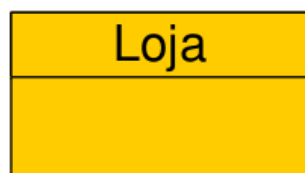


Como modelar a compra para que ela armazene uma lista de produtos com as suas quantidades?



Modelagem de sistemas

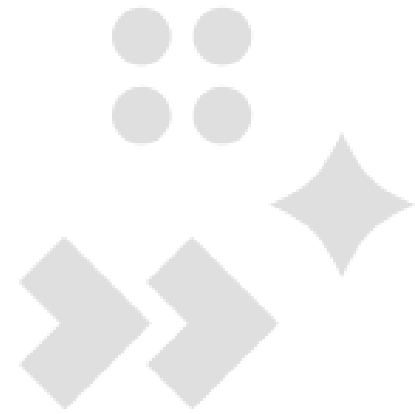
.Quais são os objetos? Definindo as classes



**Criando um novo objeto
ItemCompra que contém
um produto e uma
quantidade**



Modelagem de sistemas



.Para realizar a modelagem pense

.Quais são os objetos?

–Identificados através de substantivos

.Quais são os atributos dos objetos?

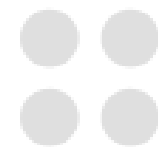
–Propriedades que o objeto deve ter

.Quais são os comportamentos (métodos) dos objetos?

–Identificados através de verbos



Modelagem de sistemas



.Deseja-se modelar e implementar uma aplicação para uma Loja Virtual. Para realizar esta tarefa você sabe que: a **Loja Virtual deverá possuir catálogo de produtos e coleção de clientes**

.Cada **produto possui código numérico, nome e preço**

.Cada **cliente possui código numérico, nome, endereço e CPF**

.A loja deverá ser capaz de listar todos produtos do catálogo, bem como todos clientes cadastrados

.Um cliente, ao fazer uma **compra, relaciona uma lista de produtos e suas respectivas quantidades**

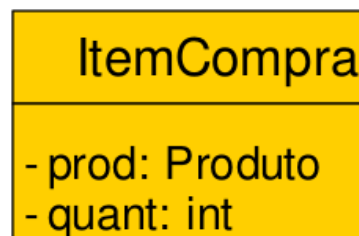
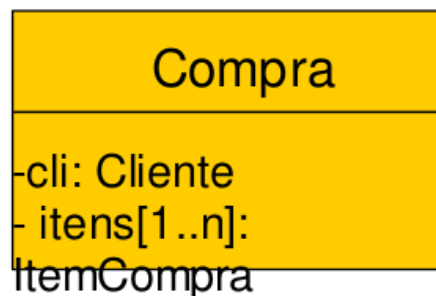
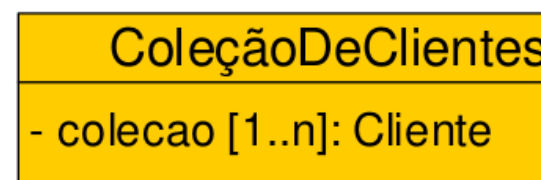
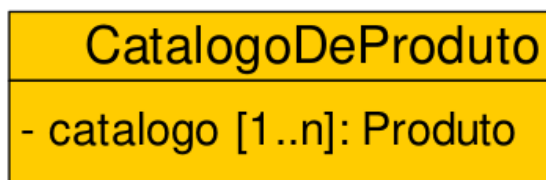
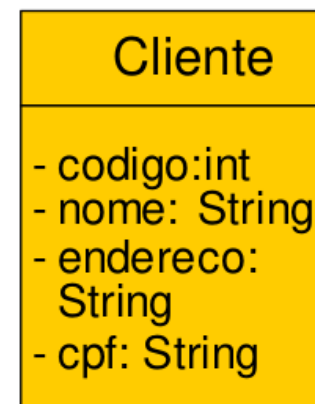
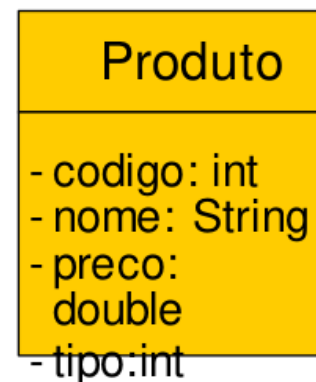
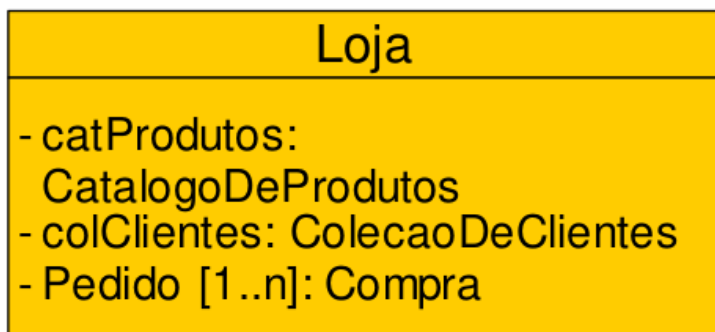
.Ao final da compra, a loja apresenta pedido completo do cliente, incluindo seu nome, lista de produtos e suas quantidades, preços unitários, e total da compra



Modelagem de sistemas



.Atributos





Modelagem de sistemas



.Para realizar a modelagem pense

.Quais são os objetos?

–Identificados através de substantivos

.Quais são os atributos dos objetos?

–Propriedades que o objeto deve ter

.Quais são os comportamentos (métodos) dos objetos?

–Identificados através de verbos



.Lembre-se do Encapsulamento!

- O acesso a uma classe deve ser realizado através de sua interface pública
- Um bom encapsulamento
 - Permite alterar a implementação interna da classe sem alterar os usuários da classe
 - Evita efeitos colaterais
 - Facilita a manutenção de código



.Observações

- Cuidado com as classes onipotentes
- As classes devem ser coesas
 - Específicas para desempenhar um papel em um contexto
 - Se as responsabilidades não são relacionadas divide-se a mesma em novas classes
- Deve-se evitar adicionar responsabilidades só porque “podem vir a ser úteis”



Modelagem de sistemas



.Deseja-se modelar e implementar uma aplicação para uma Loja Virtual. Para realizar esta tarefa você sabe que: a **Loja Virtual deverá possuir catálogo de produtos e coleção de clientes**

.Cada produto possui código numérico, nome e preço

.Cada cliente possui código numérico, nome, endereço e CPF

.A loja deverá ser capaz de listar todos produtos do catálogo, bem como todos clientes cadastrados

.Um **cliente**, ao fazer uma compra, **relaciona uma lista de produtos e suas respectivas quantidades**

.Ao final da compra, a **loja apresenta pedido completo do cliente, incluindo seu nome, lista de produtos e suas quantidades, preços unitários, e total da compra**



Modelagem de sistemas

.Métodos



Aplicação
+ main()

Loja
+ Loja() + cadastraProdutos(Produto p) + listaProdutos() + retornaProduto(int pos): Produto + cadastraCliente(Cliente c) + listaClientes() + retornaCliente(int pos): Cliente + insereCompra(Compra c) + listaCompras()

Produto
+ Produto(int c, String n, double p, int t) + Produto(Produto p) + getNome(): String + getPreco(): double + getCodigo(): int + getTipo(): int

Compra
+ Compra(Cliente c) + getCliente(): Cliente + insereItemCompra(Produto p, int q) + imprimetensCompra() + calculaTotalCompra()

ItemCompra
+ ItemCompra(Produto p, int q) + getProduto(): Produto + getQuantidade(): int

Um pouco mais elegante

