

Programação Orientada a Objetos (POO)

Programação Orientada a Objetos - Conceito

- Programação orientada a objetos é um paradigma de programação em que um sistema é construído como um conjunto de objetos que colaboram para resolver um problema computacional.
- “O paradigma da orientação a objetos visualiza um sistema de software como uma coleção de agentes interconectados chamados objetos. Cada objeto é responsável por realizar tarefas específicas. Para cumprir com algumas das tarefas sob sua responsabilidade, um objeto pode ter que interagir com outros objetos. É pela interação entre objetos que uma tarefa computacional é realizada.” [Bezerra, 2007]

Princípios da orientação a objetos

- Qualquer coisa pode ser um objeto
- Objetos realizam tarefas por meio da requisição de serviços a outros objetos
- Cada objeto pertence a uma determinada classe. Uma classe agrupa objetos similares
- A classe é um repositório para comportamento associado ao objeto
- Classes são organizadas em hierarquias

[Bezerra, 2007]

Objetos

- “ Uma unidade autônoma que contém seus próprios dados que são manipulados pelos processos definidos para o objeto e que interage com outros objetos para alcançar um objetivo” [Bezerra, 2007]
- Objetos são caracterizados por:
 - identidade;
 - atributos;
 - comportamento.
- O estado de um objeto é representado pelo conjunto de valores de seus atributos.


Exemplo de objeto

Esfera Vermelha

Objeto Esfera	
Atributos (nome, valor)	
	(peso , 200 g)
	(raio , 60 cm)
	(elasticidade , alta)
	(cor , vermelha)
Comportamento	
aumentar, diminuir, se mover	

Exemplo de objeto

Um Financiamento

Objeto Financiamento	
Atributos (nome, valor)	
	(valor, R\$ 150)
	(número de parcelas, 3)
	(percentual de juros, 1%)
Comportamento	
calcula parcela	

Classe

"Uma classe é uma representação abstrata para um conjunto de objetos que possuem estrutura e comportamento semelhantes." [Chavez]

- › Por este motivo, as classes definem:
 - os atributos que irão descrever o objeto;
 - os métodos que definem o comportamento dos mesmos.

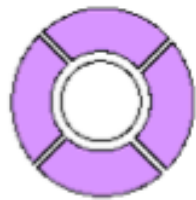


Classe

- Em programação orientada a objetos:



Atributos: dados que pertencem a cada instância da classe (objeto); são definidos sob a forma de variáveis.



Métodos: definem o comportamento do objeto; representados por módulos.


Exemplo de classe

Esfera

Classe Esfera	
Atributos (nome, tipo)	
	(peso , real)
	(raio , real)
	(elasticidade , string)
	(cor , color)
Comportamento	
aumentar, diminuir, se mover	

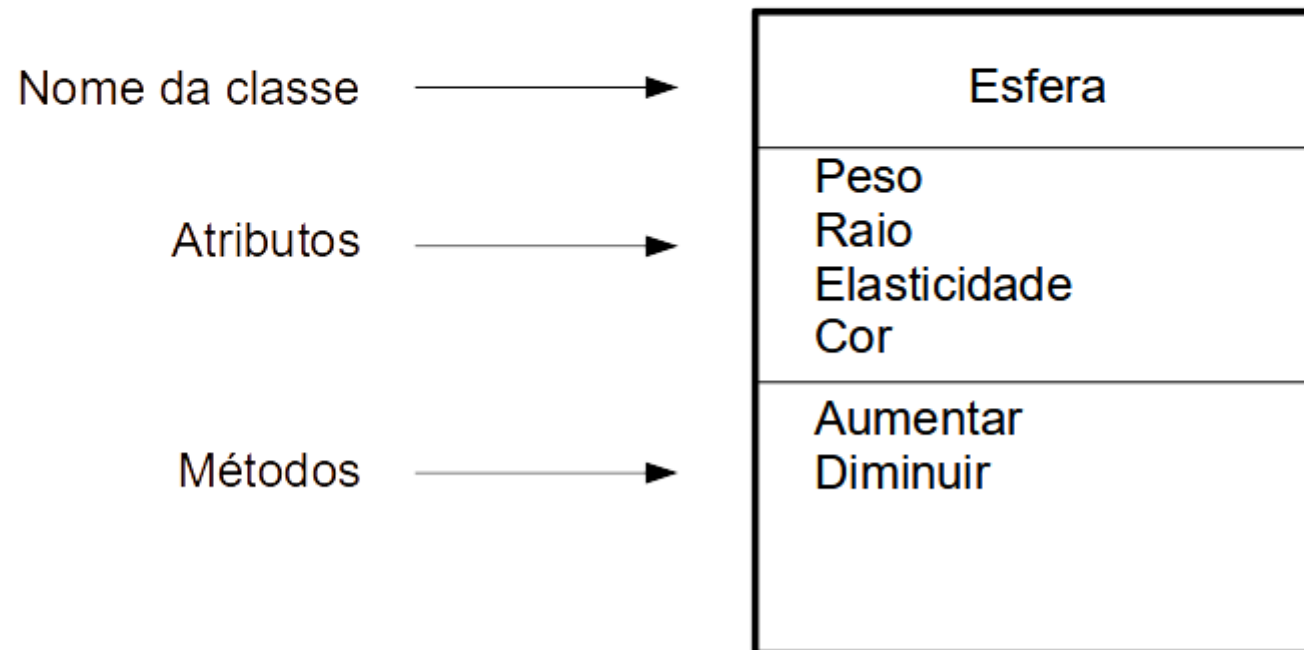
Exemplo de classe

Financiamento

Classe Financiamento	
Atributos (nome, tipo)	
	(valor , real)
	(número de parcelas , inteiro)
	(percentual de juros , real)
Comportamento	
calcula parcela	

UML – representação de uma classe

- Uma classe é representada usando-se um retângulo dividido em 3 partes



Objetos e Classes

- Objetos são instâncias de classes
- Objetos da mesma classe têm:
 - O mesmo conjunto de atributos (os valores dos atributos podem ser diferentes)
 - O mesmo conjunto de métodos

Abstração

- “A abstração é um processo mental pelo qual nós, seres humanos, nos atemos aos aspectos mais importantes (relevantes) de alguma coisa, ao mesmo tempo que ignoramos os menos importantes.” [Bezerra, 2007]
- “(...)uma abstração de algo é dependente de perspectiva (contexto)sobre a qual uma coisa é analisada: o que é importante em um contexto pode não ser importante em outro.”[Bezerra, 2007]

Encapsulamento

- › “Na programação orientada a objetos, os objetos comunicam-se entre si através de mensagens. A única coisa que um objeto conhece sobre outro objeto é a sua interface de comunicação. Os dados e a lógica de cada objeto são mantidos escondidos dos outros objetos. Em outras palavras, a interface encapsula o código e os dados do objeto”. [IBM]

Encapsulamento

Atributos Privados e Métodos

- › Na POO uma classe possui, em geral, os **atributos privados** e os **métodos podem ser públicos**, tornando o objeto como uma caixa preta onde só aparece o suficiente para que o programa possa utilizá-lo.

Polimorfismo

- É a capacidade de duas ou mais classes de objetos responderem à mesma mensagem, cada uma do seu modo.
- Exemplo:
 - Quando uma aplicação que trabalha com diversas figuras geométricas precisa desenhar esses objetos, ela não precisa conhecer as especificidades de cada figura, basta que elas possuam um método para desenhá-las que responda à mesma mensagem. A aplicação só precisa pedir que cada figura se desenhe.

Herança

- Uma classe pode estender uma classe preexistente herdando seus atributos e métodos.
- A classe herdeira é denominada subclasse e a outra é denominada superclasse.
- Esse mecanismo facilita o compartilhamento de atributos e comportamento entre um conjunto de classes semelhantes.

