

# Conceitos fundamentais da Orientação a Objetos

Prof. Felipe Scheidt

2016

# Conceitos fundamentais

- Classe
- Objeto
- Atributos
- Métodos
- Abstração
- Encapsulamento

# Classes

- Podemos ver classes como moldes para a construção de objetos
- É a descrição de um grupo de objetos com propriedades similares
- Busca-se identificar quais são as características em comum nesses objetos, tais como:
  - atributos ou propriedades
  - comportamento ou operações

# Exemplo de Classes

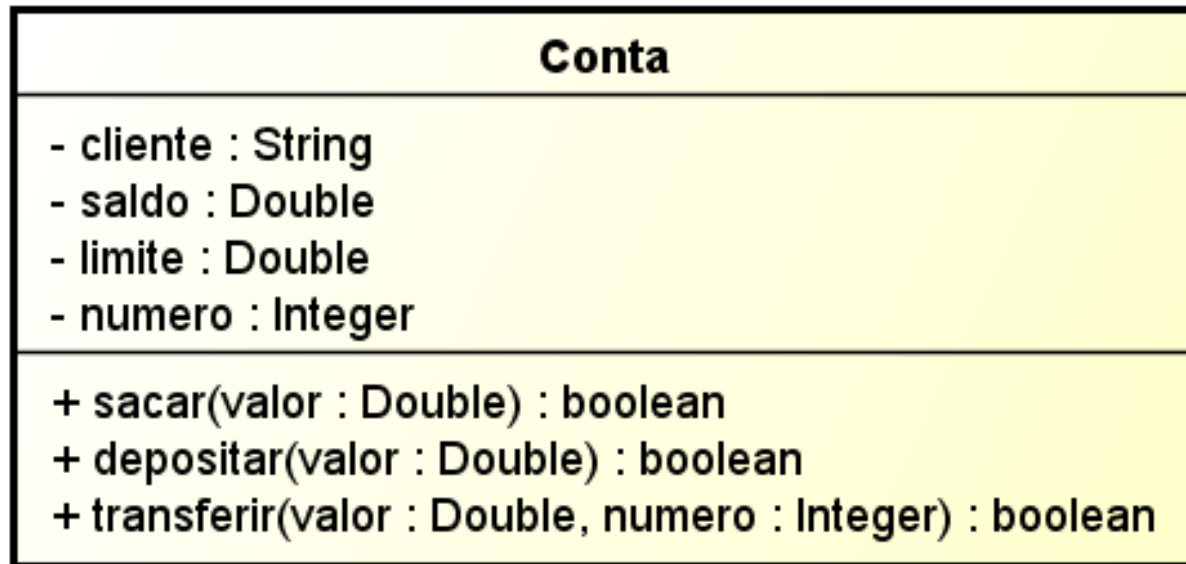
- Suponha que você precisa fazer a programação de uma conta de um banco.
- Perguntas:
  - que tipo de informações ou propriedades uma conta possui?
  - que tipo de ações podem ser realizadas numa conta?

# Classe Conta

- Propriedades:
  - nome do cliente
  - saldo
  - limite
  - número da conta
- Ações:
  - sacar um valor
  - depositar um valor
  - transferir um valor

# Modelo da Classe Conta

- Com o diagrama de classes podemos representar a classe Conta no seguinte modelo:



# + Exemplos de Classes

- Conceitos sinônimos de classe:
  - Receita
  - Planta de uma casa
  - Molde
  - Especificação
  - Conceito
  - Ideia
  - Projeto
  - Modelo

# Classe no Java

- Uma representação da Classe Conta no Java:

```
public class Conta {  
    String cliente;  
    Double saldo;  
    Double limite;  
    Integer numero;  
}
```



# Abstração

- Um importante aspecto a ser considerado durante a modelagem de uma classe é a abstração.
- Consistem em escolher as características essenciais de uma determinada entidade e descartar as outras que são irrelevantes.
- Sob outro ponto de vista, a abstração consiste em focar no que um objeto faz, buscando melhorar o entendimento sobre o que a aplicação deve fazer.

# Objetos

- Um objeto representa uma entidade que pode ser física, conceitual ou de software.
- Cada objeto é único e possui sua própria identidade.  
Não há dois objetos iguais
- É uma abstração de algo que possui fronteira definida e significado
- Um objeto passa a existir a partir de um molde. Este molde é a classe
- Para que um objeto seja usado é necessária a criação de uma referência.

# Objetos

- Um objeto é algo que possui:
  - um estado
  - uma identidade única
  - um comportamento
- Objetos possuem um estado interno, que representa seu estado atual.
- Por exemplo: uma xícara pode estar cheia, uma lâmpada pode estar ligada ou desligada, etc...

# Criando um objeto

- O operador **new()** permite instanciar um novo objeto a partir de uma classe:

```
public class Programa {  
    public static void main(String[] args) {  
        Conta c = new Conta();  
    }  
}
```

# Atributos

- Atributos representam o estado interno de um objeto.
- Nome, idade e RG são exemplos de atributos de objetos da classe Pessoa.
- Cor, peso e montadora são atributos de objetos da classe Carro.
- Cada atributo tem um valor único para cada instância de objeto.

# Métodos

- Dentro da definição da classe também são declarados os métodos.
- Métodos representam ações que um objeto pode executar (seu comportamento).
- Por exemplo, **abrir**, **salvar** e **imprimir** são operações que podem ser aplicadas a objetos da classe *Arquivo*.

# Encapsulamento

- Protege o acesso externo aos dados de um objeto
- Permite esconder ou ocultar a complexidade da implementação de um objeto.
- Vantagens?
  - evita operações ilegais
  - manutenção do código muito mais fácil, pois permite modificar um objeto sem afetar aplicações que usam este objeto.

# Encapsulamento

- Métodos *get* e *set* são formas de proteger os atributos.
- Além disso precisamos modificar a visibilidade do atributo com os modificadores de acesso.
- A prática mais comum é mudar a visibilidade dos atributos da classe para *private* e disponibilizar *getters* e *setters*.



# Exercícios

- 1) O que é uma referência? Qual a relação entre classe, objeto e referência?
- 2) Existe algum problema com o código abaixo? Se sim explique e proponha uma correção.

```
public class App {  
    String versao = "KitKat";  
    public static void main(String[] args) {  
        System.out.println(versao);  
    }  
}
```

# Exercícios

- 3) Implemente uma Classe que representa uma Data. Quais atributos são necessários nesta classe? Que métodos são necessários para garantir um bom encapsulamento? Quais regras de negócio preciso implementar para garantir consistência no valor da data? Execute o programa demonstrando o seu funcionamento.

# Exercícios

- 4) Modele um funcionário. Ele deve ter o nome do funcionário, o departamento onde trabalha, seu salário (double), a data de entrada no banco (String), seu RG (String) e um valor booleano que indique se o funcionário está na empresa no momento ou se já foi embora.
- Você deve criar alguns métodos de acordo com sua necessidade. Além deles, crie um método bonifica que aumenta o salario do funcionário de acordo com o parâmetro passado como argumento. Crie, também, um método demite, que não recebe parâmetro algum, só modifica o valor booleano indicando que o funcionário não trabalha mais aqui.
  - A idéia aqui é apenas modelar, isto é, só identifique que informações são importantes e o que um funcionário faz.

# Material de apoio

- <https://www.caelum.com.br/apostila-java-orientacao-objetos/>