

BANCO DE DADOS EM APLICATIVOS MOBILE JAVA

MÓDULO ANDROID

SENAI

Webinar 1 de 2

Msc. Luiz Carlos Machi Lozano

Agenda

- Boas-vindas e apresentação do tutor/instrutor.
- Aprofundar os temas:
 - Classes abstratas.
 - Interfaces.
- Parabenizar os alunos pelo esforço empenhado.
- Bate-papo e tira-dúvidas.

Classes

- a) Classes abstratas
- b) Interfaces

Classes abstratas

- Genéricas
- Classe modelo
- Sem instanciação

Interface

Contrato aceito por ambas as partes

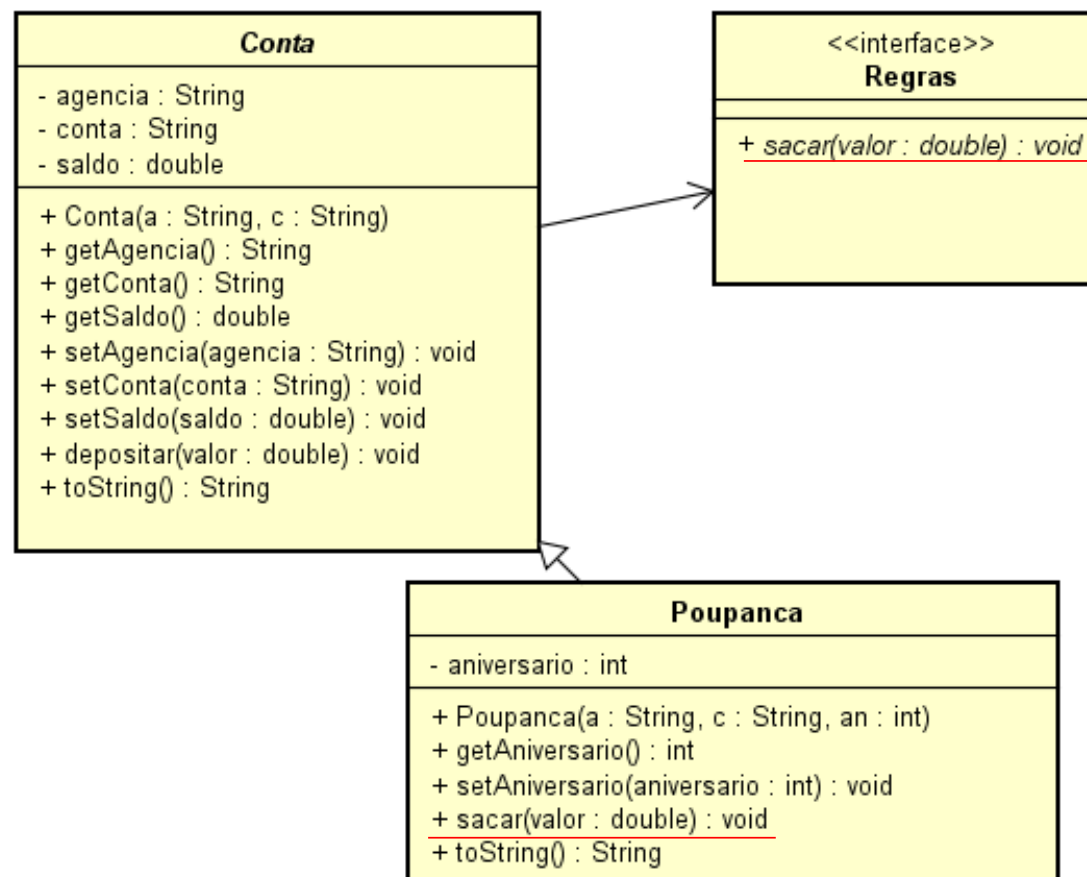
Portanto, as classes (ou grupo de classes) que herdaram sua implementação devem possuir **métodos ou propriedades em comum**.

Interface

- Dados abstratos (contêm definição, mas não implementação).
- Implementação na subclasse (palavra reservada **interface**).

a) Interface

- Exemplos



a) Implementação da interface Regras

```
public interface Regras {  
    void sacar(double valor);  
}
```

a) Implementação da classe Conta

```
public abstract class Conta {  
    private String agencia;  
    private String conta;  
    private double saldo;  
  
    public Conta(String a, String c) {  
        this.setAgencia(a);  
        this.setConta(c);  
        this.setSaldo(0);  
    }  
    public String getAgencia() { return agencia;}  
    public void setAgencia(String agencia) { this.agencia = agencia;}  
    public String getConta() { return conta;}  
    public void setConta(String conta) { this.conta = conta;}  
    public double getSaldo() { return saldo;}  
    public void setSaldo(double saldo) { this.saldo = saldo;}
```

```
public void depositar(double valor){ //método concreto  
    this.setSaldo(this.saldo + valor);  
}  
@Override  
public String toString(){  
    return "Agencia: " + this.getAgencia() +  
        "\nConta: " + this.getConta() +  
        "\nSaldo: " + this.getSaldo();  
}
```

a) Implementação da classe Poupanca

```
public class Poupanca extends Conta implements Regras{
    private int aniversario;

    public Poupanca(String a, String c, int an) {
        super(a, c);
        this.setAniversario(an);
    }
    public int getAniversario() { return aniversario;}
    public void setAniversario(int aniversario) { this.aniversario =
aniversario;}
```

```
@Override
public void sacar(double valor){
    if(valor <= super.getSaldo())
        super.setSaldo(super.getSaldo() - valor);
    else
        System.out.println("Saldo insuficiente");
}

@Override
public String toString(){
    String texto = super.toString();
    texto += "\nAniversario: " + this.getAniversario();
    return texto;
}
}
```

d) Testando as classes

- Exemplo de classe/método construtor, atributos e método implementado.

```
public class Principal {  
    public static void main(String[] args) {  
        Poupanca cpp = new Poupanca("7800","11713-1",15);  
        cpp.depositar(500);  
        cpp.sacar(501);  
        System.out.println(cpp);  
    }  
}
```

Bate-papo e tira-dúvidas

**Bons estudos e
até breve!**