



Desenvolvimento de Software para Dispositivos Móveis

Aula 3 - Java - Revisão

Professor: Anderson Almada

Tipos primitivos

TIPO	TAMANHO
boolean	1 bit
byte	1 byte
short	2 bytes
char	2 bytes
int	4 bytes
float	4 bytes
long	8 bytes
double	8 bytes

IF-ELSE

```
int idade = 15;

boolean amigoDoDono = true;

if (idade < 18 && amigoDoDono == false) {
    System.out.println("Não pode entrar");
} else {
    System.out.println("Pode entrar");
}
```

WHILE

```
int idade = 15;
while (idade < 18) {
    System.out.println(idade);
    idade = idade + 1;
}
```

FOR

```
for (int i = 0; i < 10; i = i + 1) {  
    System.out.println("olá!");  
}
```

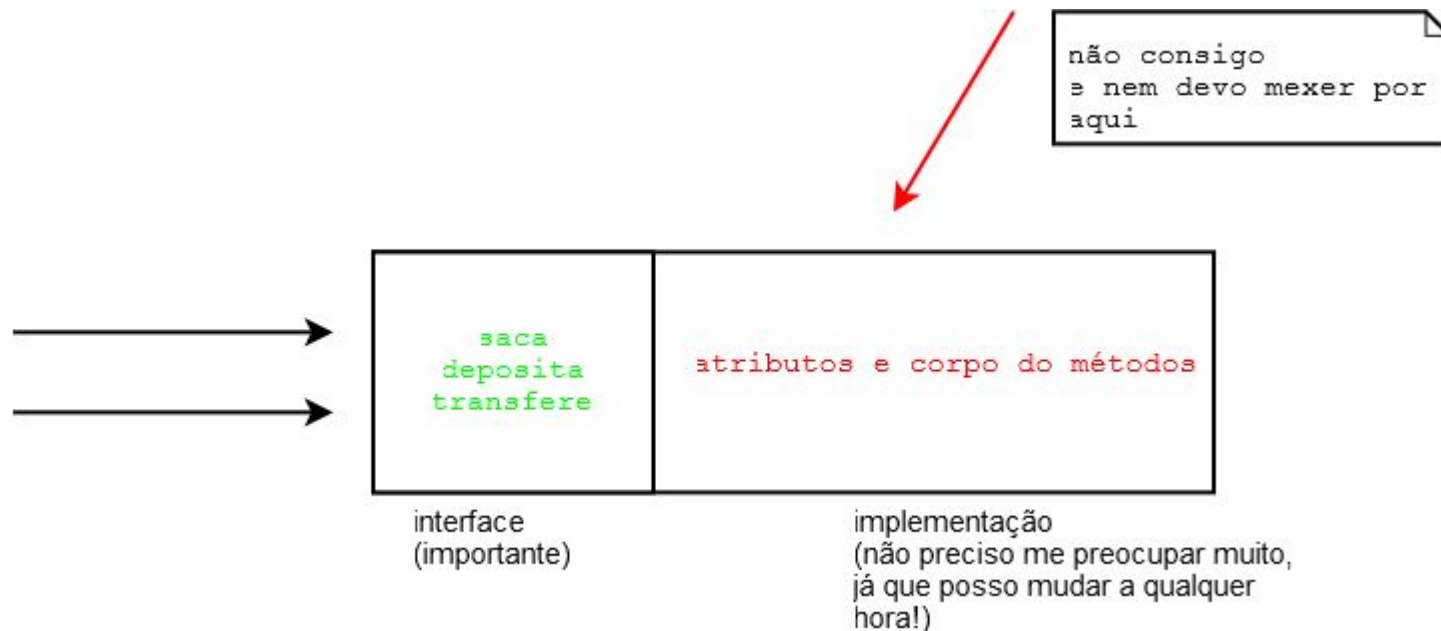
CLASS

```
class Cliente {  
    String nome;  
    String sobrenome;  
    String cpf;  
}  
  
class Conta {  
    int numero;  
    double saldo;  
    Cliente titular;  
    // ..  
}
```

CLASS

```
class Carro {  
    // Métodos  
  
    void liga() {  
        System.out.println("O carro está ligado");  
    }  
  
    void acelera(double quantidade) {  
        double velocidadeNova = this.velocidadeAtual + quantidade;  
        this.velocidadeAtual = velocidadeNova;  
    }  
    //  
}
```

Encapsulamento



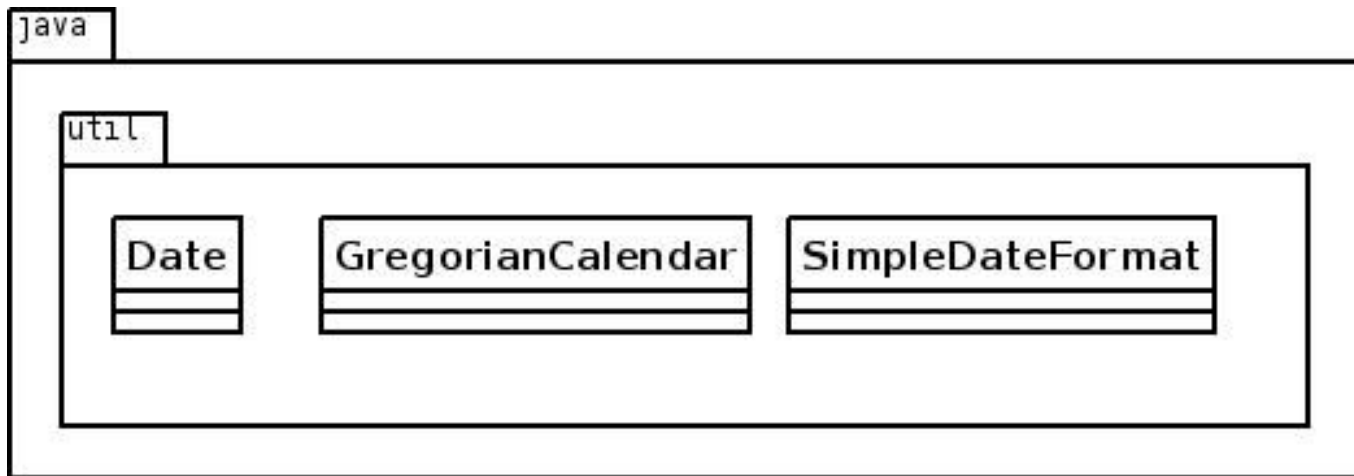
Getters e Setters

```
class Conta {  
    private String titular;  
    private double saldo;  
  
    public double getSaldo() { return this.saldo; }  
    public void setSaldo(double saldo) { this.saldo = saldo; }  
    public String getTitular() { return this.titular; }  
    public void setTitular(String titular) { this.titular = titular; }  
}
```

Construtor

```
class Conta {  
    String titular;  
    int numero;  
    double saldo;  
  
    // construtor  
    Conta(String titular) {  
        this.titular = titular;  
    }  
    // Métodos  
}
```

Pacotes



Herança

```
class Funcionario {  
    private String nome;  
    private String cpf;  
    private double salario;  
}
```

Herança

```
class Gerente extends Funcionario {  
    private int senha;  
    private int numeroDeFuncionariosGerenciados;  
    public boolean autentica(int senha) {  
        if (this.senha == senha) {  
            System.out.println("Acesso Permitido!"); return true;  
        } else {  
            System.out.println("Acesso Negado!"); return false;  
        }  
    }  
}
```

Herança - Sobrescrita

```
class Funcionario {  
    //  
    public double getBonificacao() {  
        return this.salario * 0.10;  
    }  
}
```

Herança - Sobrescrita

```
class Gerente extends Funcionario {  
    //  
    public double getBonificacao() {  
        return this.salario * 0.15;  
    }  
}
```

Polimorfismo

```
Funcionario funcionario1 = new Gerente();  
funcionario1.setSalario(5000.0);  
controle.registra(funcionario1);
```

```
Funcionario funcionario2 = new Funcionario();  
funcionario2.setSalario(1000.0);  
controle.registra(funcionario2);
```


Classe abstrata

```
public abstract class Funcionario {  
    public abstract double getBonificacao();  
    // outros atributos e métodos  
}
```

Interface

```
public interface Autenticavel {  
    boolean autentica(int senha);  
}
```

Exceções

```
try {  
    for (int i = 0; i <= 15; i++) {  
        cc.deposita(i + 1000);  
        System.out.println(cc.getSaldo());  
        if (i == 5) {  
            cc = null;  
        }  
    }  
} catch (NullPointerException e) {  
    System.out.println("erro: " + e);  
}
```

Arrays

```
Conta[] minhasContas = new Conta[10];  
minhasContas[0] = new ContaCorrente();  
minhasContas[1] = new ContaPoupanca();
```

List

```
List lista = new ArrayList();  
lista.add("Manoel");  
lista.add("Joaquim");  
lista.add("Maria");
```

```
List<ContaCorrente> contas = new ArrayList<ContaCorrente>();  
contas.add(c1);  
contas.add(c3);  
contas.add(c2);
```

Map

```
Map<String, ContaCorrente> mapaDeContas = new HashMap<>();  
mapaDeContas.put("diretor", c1);  
mapaDeContas.put("gerente", c2);
```

Link importante

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



Dúvidas??

E-mail: almada@crateus.ufc.br