ESCOLA POLITÉCNICA E DE ARTES

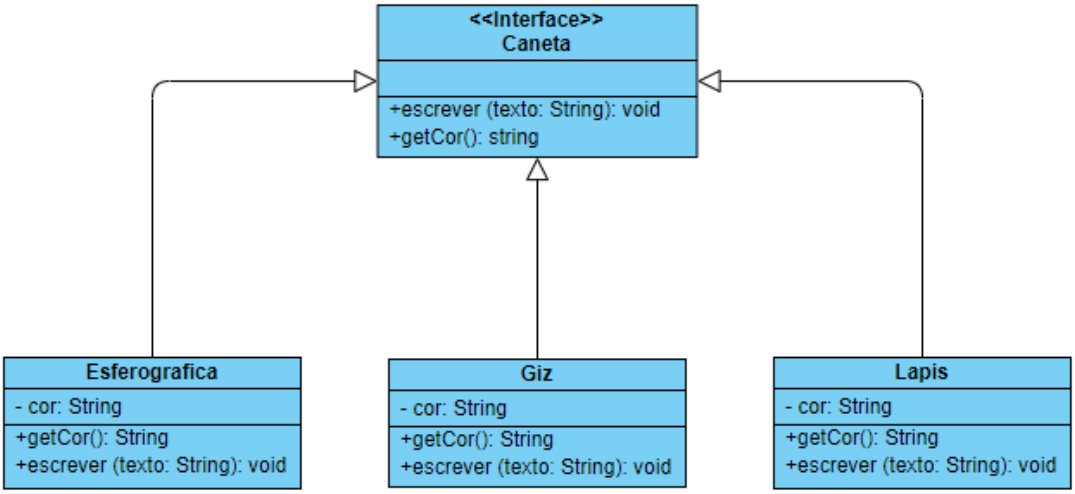
Disciplina: TECNOLOGIA DE CONSTRUÇÃO DE SOFTWARE (CMP 1069)

Prof. MSc. Anibal Santos Jukemura

# Exercícios: INTERFACES

Para cada exercício crie um novo projeto ou separe as classes por pacotes (packages). Teste todos os programas na classe principal (classe que contém o método main).

**Exercício1** – Implemente a seguinte interface:




Para cada classe, crie um construtor passando como parâmetro, uma string Cor para inicializar o atributo “cor” de cada uma.

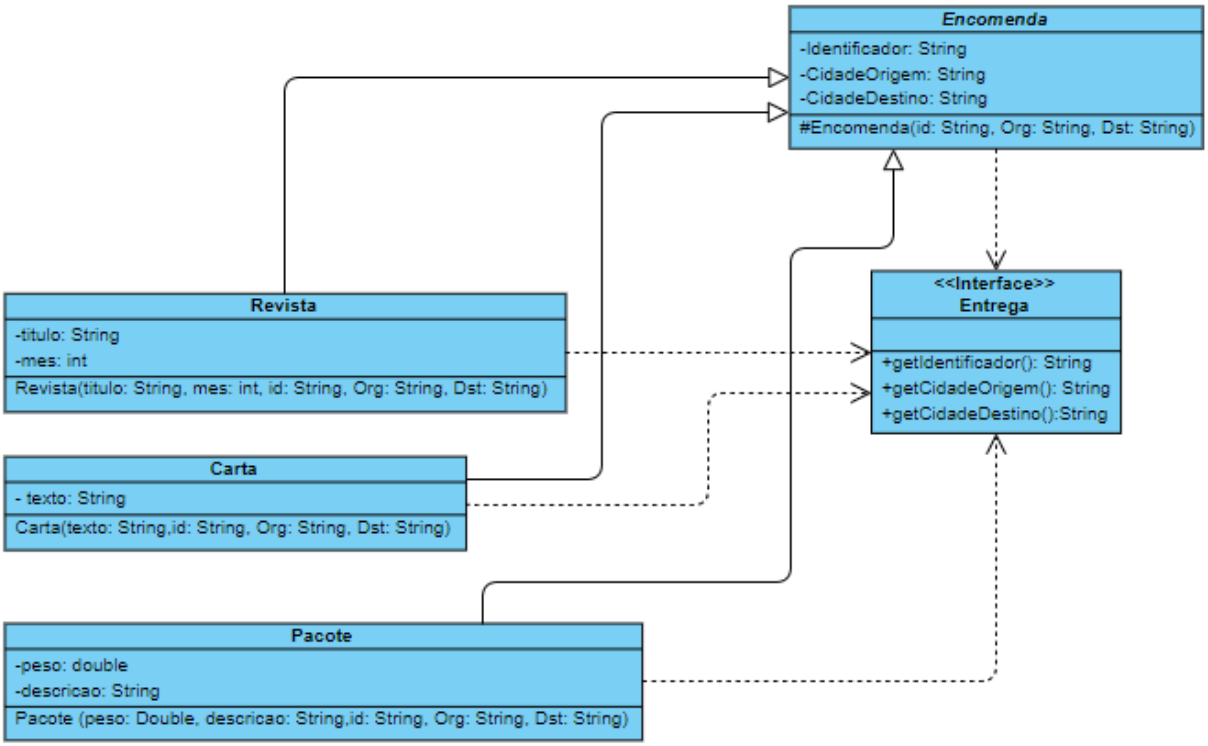
O método escrever imprime o texto:

- Para Esferográfica: “Usando “ + texto + “à caneta esferográfica.”
- Para Giz: “Usando “ + texto + “com giz”
- Para Lápis: “Usando “ + texto + “à lápis.”

Crie um método estático chamado escreverTexto com dois parâmetros: **Caneta c** e **String texto**. **Dentro desse método**, chame os métodos **escrever** de caneta e em seguida emita uma mensagem “Cor = “, seguida do método getCor(), usando System.out.Println.

No programa principal, crie uma esferográfica azul, um giz verde e um lápis preto. Para cada um deles, chame o método escreverTexto, passando como parâmetro cada um dos três objetos criados, um de cada vez, junto com o texto “Teste de Escrita”. **Perceba que nessa aplicação, a interface será utilizada para implementar o conceito de polimorfismo (uma caneta pode ser uma esferográfica, um giz ou um lápis)**.

**Exercicio 2 –** Implemente uma “agência de correios” com as seguintes características:

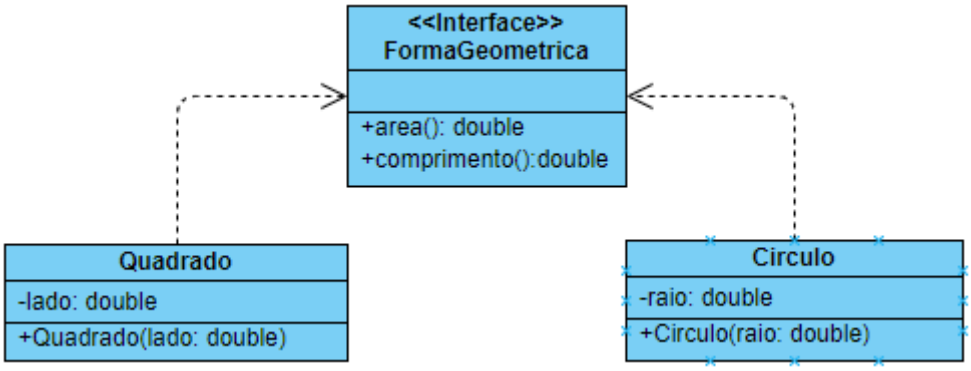



No programa principal, crie as seguintes encomendas:

| ID | Objeto | Título | Peso | Descrição | Mês | Texto | Origem | Destino |
|---|---|---|---|---|---|---|---|---|
| 1001 | Revista | Veja | | - | 05 | - | Goiânia | São Paulo |
| 1002 | Carta | - | | - | | Presente do Papai Noel | Recife | Alagoas |
| 1003 | Pacote | - | 200.00 | Talheres de Inox | | - | Rio de Janeiro | Porto Alegre |

Crie um método estático chamado entregas com um parâmetro: **Entrega e**. **Dentro desse método**, chame os métodos **para imprimir todas as** informações de cada encomenda. Perceba que, para usarmos o polimorfismo aqui, precisamos identificar o tipo do objeto, já que cada um tem uma característica específica. Considere o uso do método **instanceof**.

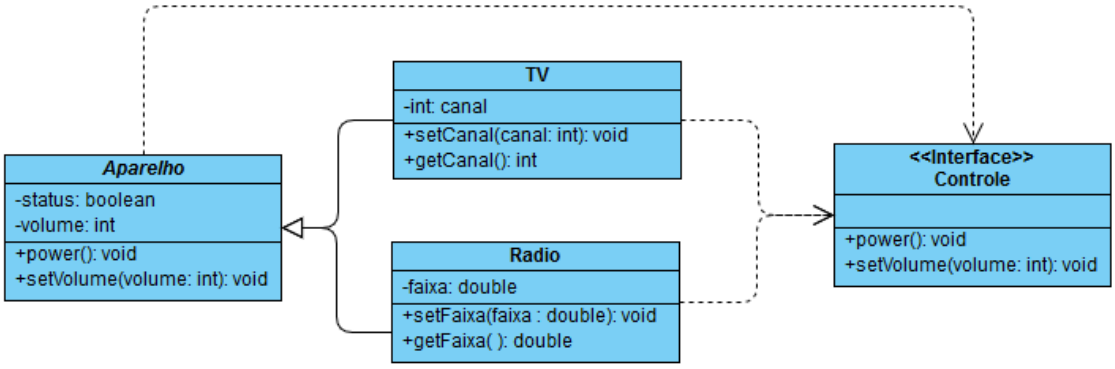
**Exercicio 3 –** Implemente a seguinte interface:




No programa principal, crie uma FormaGeometrica quadrado com lado 4 e outra círculo com raio 3.5.

Crie um método estático chamado calculo com um parâmetro: **FormaGeometrica FG**. **Dentro desse método**, chame os métodos **para imprimir todas as** informações de cada figura geométrica. Perceba que, para usarmos o polimorfismo aqui, precisamos identificar o tipo do objeto, já que cada um tem uma característica específica. Considere o uso do método **instanceof**.

**Exercício 4 –** Implemente a seguinte aplicação em JAVA:




No programa principal, crie um objeto chamado Toshiba do tipo Aparelho, implementado como uma TV. Crie outro objeto chamado Pioneer do tipo Aparelho, implementado como um Radio. Implemente o seguinte MENU:

```
-----Escolha a opção---------
- 1-Liga/Desliga             -
- 2-Controla Volume          -
- 3- TV- Seleciona Canal     -
- 4- Radio: Seleciona Faixa  -
- 5- Informacoes TV          -
- 6- Informacoes Radio       -
- 0- Sair                    -
-----------------------------
Opcao:
```

**Para cada uma das opções, crie um método estático, passando como parâmetro, um tipo Controle "t"**. Considere o uso de instanceof, para implementar cada um dos métodos. Considere:

1 – Liga/Desliga: imprime o status da TV ou do Rádio (true = ON e false=OFF)

2- Controla Volume: peça ao usuário qual aparelho desejado: TV ou Rádio. Em seguida, peça  o novo valor para o volume da TV ou Rádio e configure esse novo valor no atributo volume.

3 – Peça ao usuário qual canal ele deseja e configure esse canal no atributo canal da TV.

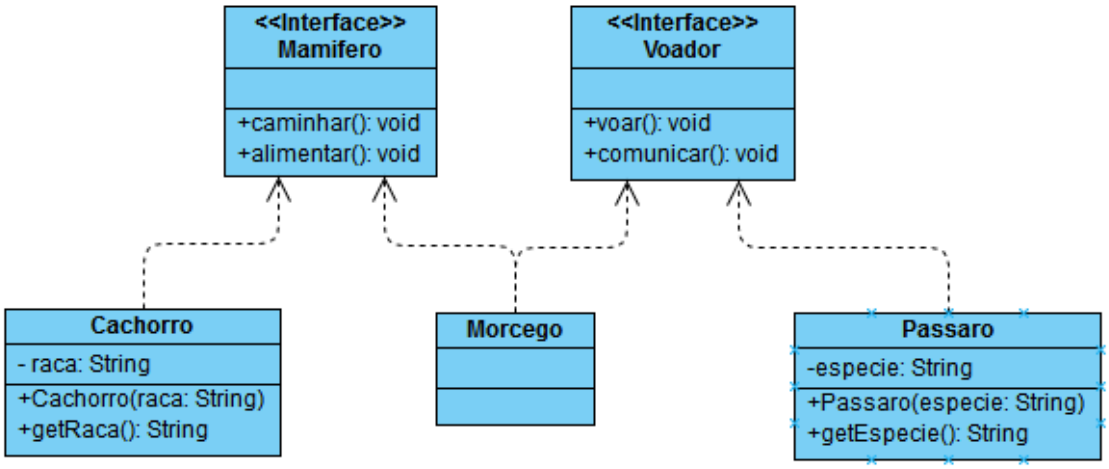
4 – Peça ao usuário qual faixa ele deseja e configure-a no atributo faixa do Radio.

5 – Imprima as informações da TV: status, canal atual, volume atual.

6 – Imprima as informações do Radio: status, faixa atual, volume atual.

Para os itens 5 e 6, considere a possiblidade de usar somente um método estático para imprimir os dados, diferenciando o parâmetro Controle através do uso de instanceof.

**Exercício 5 –** Simulação de Herança Múltipla. Implemente a seguinte aplicação em JAVA:




Perceba que em UML, os métodos que devem ser implementados nas classes podem ser omitidos, pois estão implícitos na relação.

No programa principal, crie um objeto para cachorro, um para pássaro e um para morcego e imprima todos os dados de cada um.

A classe Morcego seria um exemplo mais próximo possível de Herança Múltipla em Java.