

# Programação Orientada a Objetos - POOS3

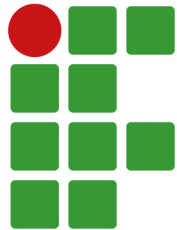
1

---

Tecnologia em Análise e Desenvolvimento de Sistemas

**Aula 6**  
Interfaces

2º semestre de 2018



**INSTITUTO FEDERAL**

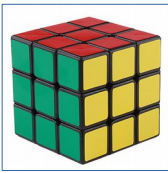
São Paulo

Câmpus Araraquara

# Momento de revisão

---





# Problema

- Cliente deseja implementar um sistema para controle de uma fila bancária para atendimento dos clientes.
  - O sistema de controle de fila de um banco opera da seguinte forma:
    - O cliente que chega a agência retira uma senha que pode ter prioridade preferencial (para gestantes, portadores de necessidades especiais e maiores de 65 anos) ou não preferencial.
    - A agência dispõe de um caixa preferencial e vários caixas não preferenciais. Ao decidir qual o próximo cliente a ser atendido, são consideradas duas políticas:
  - O caixa preferencial seleciona o primeiro cliente preferencial da fila. Não havendo clientes preferenciais, é selecionado o primeiro cliente da fila;
  - O caixa não preferencial respeita a ordem de chegada e seleciona o primeiro cliente que está aguardando na fila, preferencial ou não.

# Pensando...

Antes de pensar em código, o que é uma fila?



Fila é uma estrutura de dados ordenada pela entrada na qual o primeiro elemento a entrar é o primeiro elemento a sair.



Como você implementaria uma fila?



Como posso ter uma fila quando existe a questão da prioridade?



# Fila

- Implementação de fila estática.



# Interfaces

---

- Uma interface é sintaticamente semelhante a uma classe abstrata no fato de podermos especificar um ou mais métodos sem corpo.
- Esses métodos devem ser implementados por uma classe para que suas ações sejam definidas.
- Logo uma interface **especifica o que deve ser feito, mas não como deve ser feito.**
- Quando uma interface é definida, não existe um número de classes que podem implementá-la (implements), além disso, uma classe pode implementar qualquer número de interfaces.

# Interface x Classe Abstrata

## • Interfaces:

- Uma interface não é considerada uma Classe e sim uma Entidade.
- Não possui implementação, apenas assinatura, ou seja, apenas a definição dos seus métodos sem o corpo.
- Todos os métodos são abstratos.
- Seus métodos são implicitamente Públicos e Abstratos.
- Não há como fazer uma instância de uma Interface e nem como criar um Construtor.
- Funcionam como um tipo de "contrato", onde são especificados os atributos, métodos e funções que as classes que implementem essa interface são obrigadas a implementar.
- Já que Java não suporta Heranças Múltiplas, as Interfaces são usadas para implementá-las.

## • Classes Abstratas:

- As classes abstratas devem conter pelo menos um método abstrato, que não tem corpo.
- É um tipo especial de classe que não há como criar instâncias dela.
- É usada apenas para ser herdada, funciona como uma super classe.
- Uma grande vantagem é que força a hierarquia para todas as subclasses.
- É um tipo de contrato que faz com que as subclasses contemplem as mesmas hierarquias e/ou padrões.

# Material Adicional

- **Leitura Obrigatória**

- Winder, R.; Roberts, G. Desenvolvendo Software em Java. 3 ed. Rio de Janeiro: LTC, 2009.
  - Capítulos 7.6
- <https://www.devmedia.com.br/interfaces-x-classes-abstratas/13337>
- <https://www.devmedia.com.br/forum/duvidas-simples-de-certificacao-java-diferenca-entre-classe-abstrata-e-interface/566376>
- <https://www.caelum.com.br/apostila-java-orientacao-objetos/classes-abstratas/#null>
- <https://www.caelum.com.br/apostila-java-orientacao-objetos/interfaces/#interfaces>

