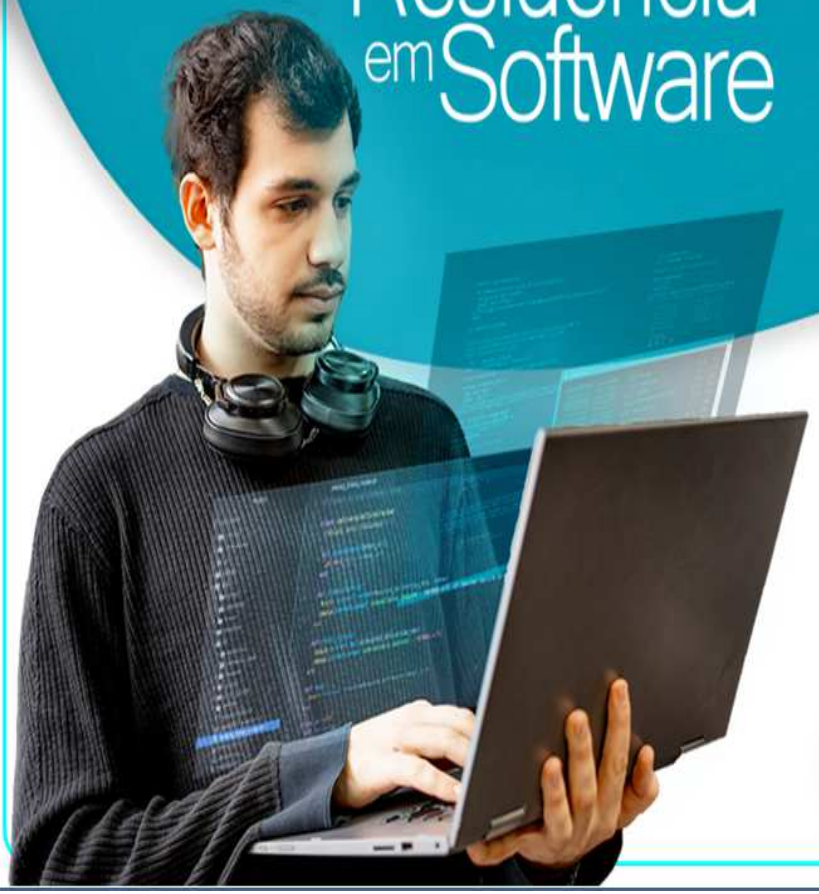
Residência em Software

Residência em Tecnologia da Informação e Comunicação

Interfaces e Classes Abstratas

Professor:

Alvaro Degas Coelho



Java

INSTITUIÇÃO EXECUTORA

cepedi UESC

COORDENADORA

MCTI FUTURO Softex

APOIO

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Contexto


Forno
Pizza
Descongelar
Tostar
Rotisseria
Desidratar
Relógio
Timer
Temp
Lâmpada
Ligar

Interfaces

- Relembrando conceitos: polimorfismo
  - Uma invocação leva a diferentes implementações
- Lembram do exemplo DataFruta
  - Diferentes classes implementam o mesmo "método"
    - Média, Mediana, Máximo, Mínimo, etc.
  - Cada um no seu contexto
    - Números em ponto flutuante, Inteiros, Datas, Nomes, etc.

Residência em Software

Um novo olhar

- Podemos ver o método de cada classe com uma mesma “função”
  - Mas executada em diferentes contextos
- Pense num método “Locomover” em diferentes animais.
  - Cobra, Borboleta, Jegue, Peixe
- Mesma “função”, execuções diferentes

Residência em Software

Interface

- Uma maneira de formalizar o Polimorfismo
  - Métodos
- Dizer: “Algumas classes terão estas funcionalidades obrigatoriamente”
  - Compartilham uma mesma interface
- Importante: define um conjunto mínimo. Não proíbe que outros métodos possam existir

Interfaces: porque

- Polimorfismo
- Reusabilidade e Componentização
  - Diferentes componentes podem “aderir” ao sistema
    - Dois objetos diferentes para executar uma busca numa coleção. Um faz a busca binária e o outro faz a busca sequencial
    - Instancia-se o objeto correto de acordo com o contexto

Residência em Software

Interfaces: como

- Uma interface define um padrão para alguns métodos. Logo
  - Não possui implementação
  - Não possui atributos
- A rigor define apenas a assinatura dos métodos
  - Ex. Supondo que exista uma interface chamada “buscador”
    - public int busca(List lista);

Residência em Software

Classes e Interfaces

- Classes **implementam** interfaces
- Por exemplo
  - class buscadorSequencial implements buscador
  - class buscadorBinario implements buscador
- As classes que implementam uma interface ficam **obrigadas** a conter os métodos da interface

De volta ao Contexto

```
interface Aquecedor {
        void ligar();
        void temorizador();
        void potencia();
        void parar();
}
```

```
class AirFries implements
Aquecedor {
        void ligar(){...};
        void temorizador() {...};
        void potencia() {...};
        void parar() {...};
}
```

```
class Microondas implements
Aquecedor {
        void ligar(){...};
        void temorizador() {...};
        void potencia() {...};
        void parar() {...};
}
```

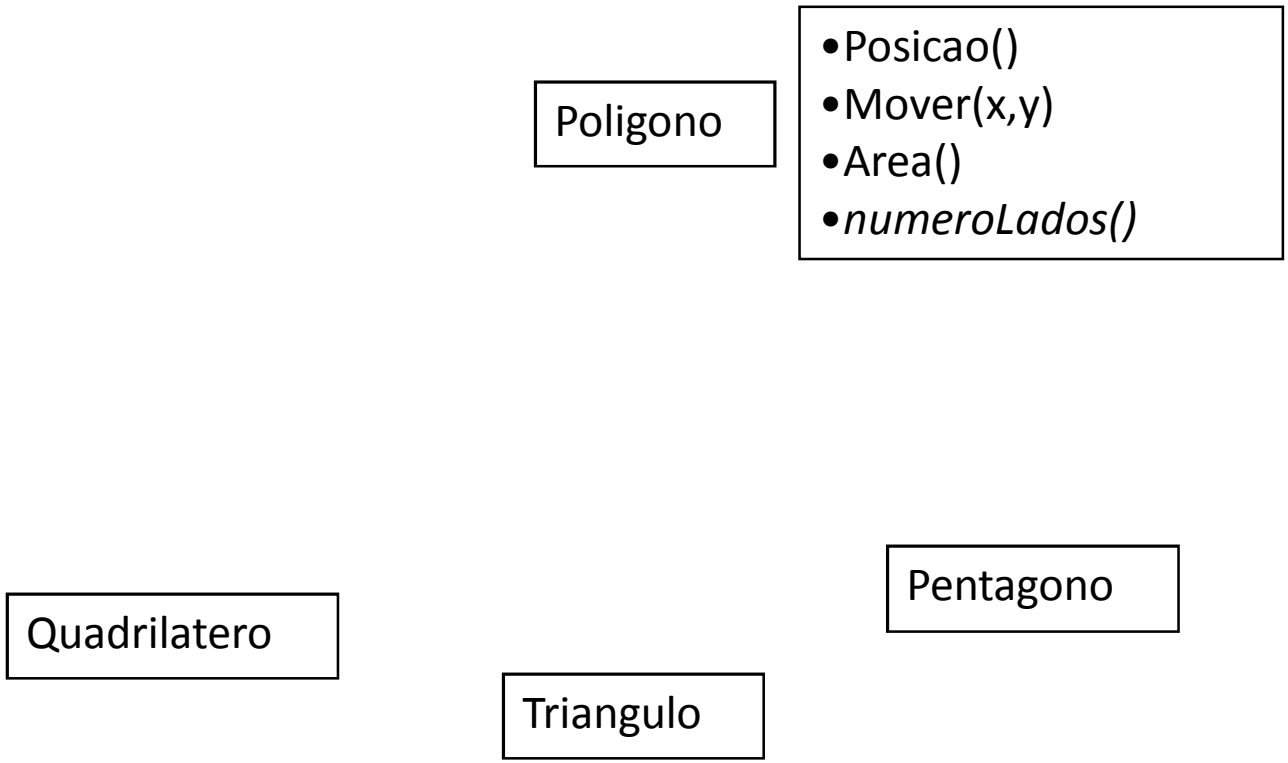
Exercício

- Crie uma interface “DadosEstatisticos”
  - Métodos: máximo, mínimo, ordenar, buscar
- Crie classes que implementam “DadosEstatísticos”
  - DatasDeNascimento, TemperaturasDoPeriodo, CidadesDoBrasil
- Crie um programa com uma lista de objetos do tipo “DadosEstatisticos”
  - Inclua diferentes objetos de diferentes classes
  - Execute os métodos

Classe Abstrata

- É uma classe que não instancia objetos
- Possui ao menos um método abstrato
  - Estes precisarão serem implementados nas subclasses
- Não é permitido se instanciar um objeto do tipo de uma classe abstrata

Exemplo


Poligono
•Posicao()
•Mover(x,y)
•Area()
•*numeroLados()*
Quadrilatero
Triangulo
Pentagono

Exemplo

```
abstract class Poligono {
        void mover(int x,int y){...}
        double area(){...}
        int numeroLados();
}
```

```
class Quadrilatero extends Poligono{
        int numeroLados(){return 4;}
        ...
}
```

Residência em Software

Exercício

- Crie uma Classe Abstrata “DadosEstatisticos”
  - Métodos: máximo, mínimo, ordenar, buscar
- Crie classes que estendem “DadosEstatísticos”
  - DatasDeNascimento, TemperaturasDoPeriodo, CidadesDoBrasil
- Crie um programa com uma lista de objetos do tipo “DadosEstatisticos”
  - Inclua diferentes objetos de diferentes classes
  - Execute os métodos

Classes Abstratas ou Interfaces

- Considere sua prioridade
  - Herança de Código?
  - Atributos comuns?
- Ou, por outro lado
  - Métodos sempre específicos
  - Herança múltipla
- SIM!
  - Java não tem herança múltipla
- Mas tem Interface Múltipla!