

Interface

- Uma interface estabelece um contrato com as classes que a implementam, obrigando-as implementar todos os métodos abstratos;
- Interface em Java pode conter variáveis de instância final (constantes) e assinatura de métodos (métodos abstratos);
- Quando uma classe não sobrescreve um ou mais métodos abstratos da interface, esta deverá ser definida como classe abstrata;
- Interface promove herança de tipo e não herança de implementação;
- Em Java uma classe pode implementar N interfaces.

Interface

Três maneiras de definir elementos de uma Interface

```
package negocio;  
public interface INota {  
  
    public double MENOR_NOTA = 0.0;  
    public double MAIOR_NOTA = 10.0;  
  
    public abstract double calcularMedia();  
}
```

```
package negocio;  
public interface INota {  
  
    public double MENOR_NOTA = 0.0;  
    public double MAIOR_NOTA = 10.0;  
  
    public double calcularMedia();  
}
```

```
package negocio;  
public interface INota {  
  
    double MENOR_NOTA = 0.0;  
    double MAIOR_NOTA = 10.0;  
  
    double calcularMedia();  
}
```

Interface

```

package negocio;
public class Nota1 implements INota {

    private double n1;
    private double n2;

    public Nota1(double n1, double n2) {
        this.setN1(n1);
        this.setN2(n2);
    }
    @Override
    public double calcularMedia() {
        return (this.n1 + this.n2) / 2;
    }
    public double getN1() { ...3 linhas }
    public void setN1(double n1) {
        if (n1 >= INota.MENOR_NOTA && n1 <= INota.MAIOR_NOTA) {
            this.n1 = n1;
        }
    }
    public double getN2() { ...3 linhas }
    public void setN2(double n2) { ...5 linhas }
}

```

Nota1 implementa/realiza a interface INota;

Para Nota1 ser concreta deve sobrescrever o método calcularMedia();

Interface

```
package aplicacao;  
import negocio.INota;  
import negocio.Nota1;  
public class Aplicacao01 {  
  
    public static void main(String[] args) {  
  
        INota nota1 = new Nota1(6, 9);  
  
        System.out.println("Media nota1 = "+nota1.calcularMedia());  
    }  
}
```

Variável nota1 é baseada no tipo INota;

A variável nota1 recebe uma instância da classe Nota1;

Interface

```
package negocio;

public class Nota2 extends Nota1 {

    public Nota2(double n1, double n2) {
        super(n1, n2);
    }

    @Override
    public double calcularMedia() {
        return (this.getN1() * 2 + this.getN2() * 3) / 5;
    }
}
```

Nota2 estende a classe Nota1;


Nota2 é do tipo Nota1 e também do tipo Inota;

Interface

```
package aplicacao;
import negocio.INota;
import negocio.Nota1;
import negocio.Nota2;
public class Aplicacao02 {

    public static void main(String[] args) {
        INota nota1 = new Nota1(6, 9);
        INota nota2 = new Nota2(6, 9);

        System.out.println("Media nota1 = "+nota1.calcularMedia());
        System.out.println("Media nota2 = "+nota2.calcularMedia());

        nota1 = nota2; 
        System.out.println("Media nota1 = "+nota1.calcularMedia());
    }
}
```

Interface

```
package negocio;  
  
public interface INotaLista extends INota{  
    void adicionaNota(double nota);  
  
}
```

INotaLista estende a interface INota;

Herança entre interfaces promovem herança de tipo (métodos abstratos e constantes;

INotaLista é do tipo INota;

Interface

```
package negocio;
import java.util.ArrayList;
public class Nota3 implements INotaLista {

    private ArrayList<Double> listaNotas;

    public Nota3() {
        this.listaNotas = new ArrayList();
    }

    public void adicionaNota(double nota) {
        this.listaNotas.add(nota);
    }

    @Override
    public double calcularMedia() {
        double soma = 0;
        for(Double nota : this.listaNotas){
            soma += nota.floatValue();
        }
        return soma / this.listaNotas.size();
    }
}
```

Nota3 implementa a interface INotaLista;

Nota3 é do tipo INotaLista;

Nota3 é do tipo INota;

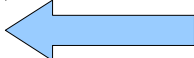
Interface

```
package aplicacao;
import ...5 linhas
public class Aplicacao03 {
    public static void main(String[] args) {

        INota nota1 = new Nota1(6, 9);
        INota nota2 = new Nota2(6, 9);
        INotaLista nota3 = new Nota3();
        nota3.adicionaNota(8);
        nota3.adicionaNota(9);

        System.out.println("Media nota1 = "+nota1.calcularMedia());
        System.out.println("Media nota2 = "+nota2.calcularMedia());
        System.out.println("Media nota3 = "+nota3.calcularMedia());

        nota1 = nota2;
        System.out.println("Media nota1 = "+nota1.calcularMedia());
        nota1 = nota3;
        System.out.println("Media nota1 = "+nota1.calcularMedia());
        //nota1.adicionaNota(7);
    }
}
```



Interface

- A classe define ambos, um tipo e uma implementação desse tipo
- O tipo define apenas a interface oferecida para acessar objetos da classe
- Um objeto pode ter muitos tipos
- Classes diferentes podem implementar o mesmo tipo