

Mas a vereda dos justos é como a luz da aurora,  
que vai brilhando mais e mais até ser dia  
perfeito. Provérbios 4:18

Engenharia de Computação

# LINGUAGEM DE PROGRAMAÇÃO ORIENTADA A OBJETOS

- ▶ Prof: José Antonio Gonçalves
- ▶ [zag655@gmail.com](mailto:zag655@gmail.com)

## Nestes Slides:

- **Orientação a Objetos em Java:**
- Conceitos sobre Herança Múltipla;
- Conceito e aplicação da estrutura de dados Interface.

# Interface

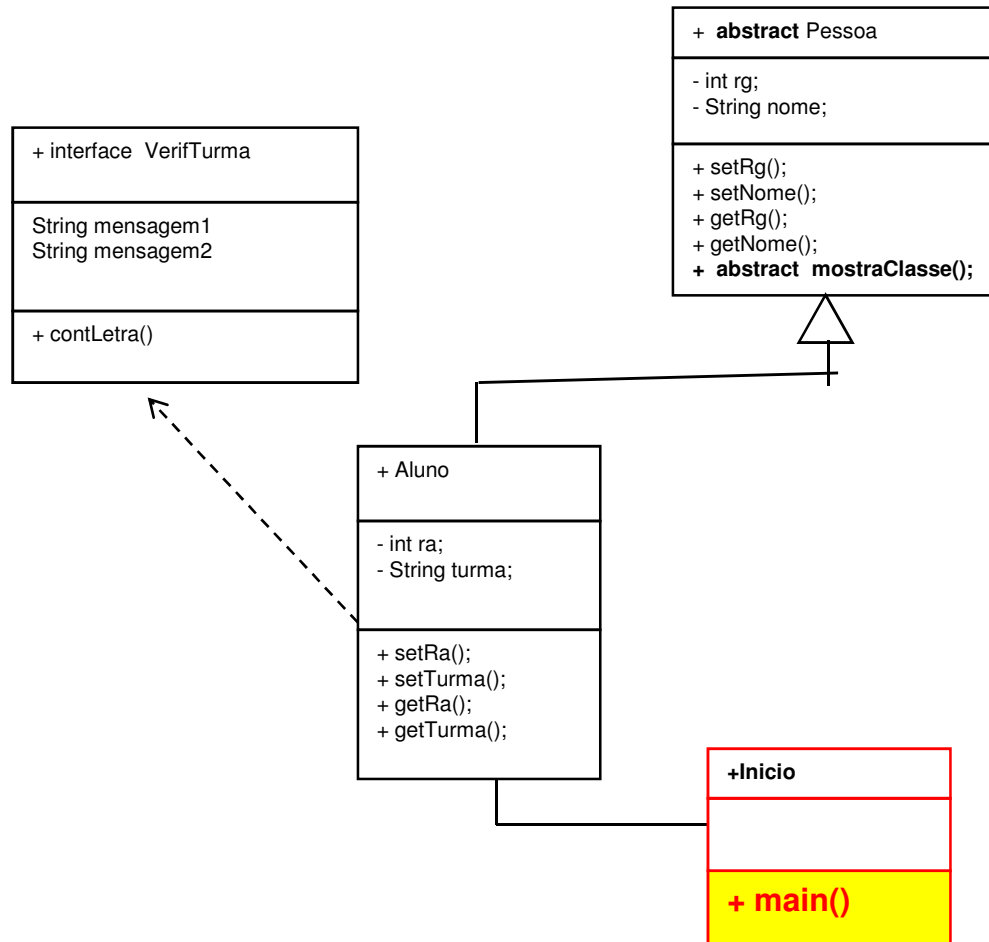
## Interface (*definição*)

**Interface:** Assim como uma classe também trata-se de um tipo de um tipo abstrato de dados:

- Porém todos os **métodos** que ela contiver **deverão ser construídos** nas classes que implementarem esta Interface, logo, em sua forma de uso, assemelha-se aos métodos abstratos;
- Caso a Interface tenha algum atributo, este será do tipo constante, isto é, não poderá ter seu valor alterado. Se comportarão como constantes (atributos “finais”);
- É utilizada para suprir a necessidade **herança múltipla**, já que não é possível implementar esta forma de herança em Java.

## Interface (*subsídios para construção*)

Observe o diagrama a seguir. Atente para a Interface associada a classe Aluno:



## Interface (código)

```
public interface VerifTurma{  
  
    String mensagem1 = "Sim";  
    String mensagem2 = "Nao";  
  
    public void contLetra();  
  
}
```

+ interface VerifTurma
String mensagem1 String mensagem2
+ contLetra()

## Interface : Aluno implementa

```
public class Aluno extends Pessoa implements VerifTurma{  
    private int ra;  
    private String turma;
```

```
    //deve colocar os métodos setters e getters
```

```
    public void mostraClasse(){  
        System.out.println("\n Estou na classe Aluno");  
    }  
    public void mostraMae(){  
        super.mostraClasse();  
    }  
}
```

Método da interface que foi implementado na classe Aluno

```
public void contLetra(){
```

```
    //mensagem1="teste";//"descomente" para testar a "constante"
```

```
    if(turma.equalsIgnoreCase("a")){  
        System.out.println("\n Turma eh A --> "+mensagem1);  
    }  
    else System.out.println("\n Turma e B --> "+mensagem2);  
}
```

```
}
```



## TESTANDO : Interface (código)

```
public class Inicio{  
    public static void main(String arg[]){  
        Aluno a = new Aluno();  
        a.setRg(50);  
        a.setNome("amor");  
        a.setRa(1);  
        a.setTurma("A");  
        System.out.println("\n RG: "+a.getRg());  
        System.out.println("\n NOME: "+a.getNome());  
        System.out.println("\n RG: "+a.getRa());  
        System.out.println("\n NOME: "+a.getTurma());  
        a.mostraClasse();  
        a.mostraMae();  
        a.contLetra();  
    }  
}
```

Chamando, através do objeto “a” (Aluno), o método da Interface que foi implementado na classe Aluno