

# Polimorfismo

## Encapsulamento e Heranças

Polimorfismo < 1 - chamada a função  
2 - orientação a objetos

① É a capacidade de se distinguir uma função de outra pela sua "assinatura" ao invés de pelo nome

Ex

```
int soma(int a, int b);  
float soma(float a, float b);  
float soma(int a);
```

O polimorfismo garante que não haverá confusão de nomes

pois o que difere ao nome ser o mesmo é a "assinatura" da função, ou seja, seu tipo e seus parâmetros, além do nome.

② É a capacidade de atribuir um valor de um tipo a uma variável que possui um supertipo compatível

Ex class: Pessoa / class Funcionário extends Pessoa

Pessoa agente = new Funcionário(...);

Lo Como Funcionário é uma Pessoa, essa declaração é válida. Isso é o polimorfismo

Nesse caso o comportamento é definido pela classe "filha", mas a ass, o método são os mesmos, inclusive com a mesma assinatura.

# Métodos e Atributos Estáticos

São métodos e atributos que pertencem à classe como um todo e não são específicos de cada objeto, sendo ao contrário, compartilhados por todos os objetos daquela classe.

Ex:

```
public class Geral {  
    static int qtd = 0;
```

```
    public Geral() {
```

```
        qtd++;
```

```
    }  
    public static int quantosTem() {
```

```
        return qtd;  
    }  
}
```

esse método não precisa ser estático

```
public class Testa Geral {  
    public static void main (String args[]) {  
        Geral g, h;  
        g = new Geral();  
        h = new Geral();  
        System.out.println (g.quantoten());  
        System.out.println (Geral.quantoten());  
    }  
}
```

Porim  
se for  
estático  
↳

0 are escape?

2

## Constantes

A linguagem Java define constantes on seguinte forma

Ex1 `public static final int piatus=4;`

Ex2 `public class Math {  
 public static final double PI=3.14159;`

`public static double seno(double rad);`

↳ Exemplo de uso de métodos estático

`c = Math.PI * r`

`area = lado * Math.seno(90°) / 2`

# É o main?

É uma convenção da linguagem Java para definir o ponto de partida para a execução de programa.

Como esse de partida é único e bem definido, além do fato de tudo em Java estar dentro das classes, faz todo o sentido o método main ser estático

---

Uma regra fundamental é que nunca se deve chamar métodos ou acessar atributos não-estáticos dentro de métodos estáticos.

Porém é possível criar um objeto do tipo da classe dentro de um método estático e usá-lo normalmente

Ex

```
public class ProgramVisual {
```

```
    private Janela janela
```

↳ atributo  
↳ classe

```
    public static void main(...) {
```

```
        janela = new Janela(); × proibido
```

Solução

```
        ProgramVisual pv = new ProgramVisual(...)
```

```
        pv.janela = new Janela(); ✓ isso é permitido
```