

**Artigo:** Herança na Programação Orientada a Objetos utilizando JAVA

**Turma:** 2º Ano Informática

**Professor:** Luiz Cesar

**Dupla:** Pedro Víctor Rocha Camargo e Joyce Kelly De Lima

Com ajuda de alguns sites a dupla pode perceber que a herança é um princípio da POO que autoriza a criação de novas classes a partir de outras anteriormente criadas. Essas novas classes são chamadas de subclasses, e as classes presentes, que deram origem às subclasses, são chamadas de superclasses, ou classes base. Uma subclasse herda métodos e atributos de sua superclasse, ou seja, a classe principal que deu origem às demais. Apesar disso, pode-se escrevê-los novamente para uma forma mais específica de representar o comportamento do método herdado.

Cada construtor das novas classes possui uma chamada super essa palavra representa uma chamada de método ou acesso a um atributo da superclasse, por isso tem esse nome. Em alguns casos pode usá-lo para invocar o construtor da superclasse. ( Exemplo abaixo)

```
public class main {  
    public static void main(String[] args) {  
        Aluno i = new Aluno("Jose Francisco", "123.456.789-00", new Date());  
        System.out.println("Veja como os atributos foram preenchidos\n\nNome: " +  
i.nome);  
        System.out.println("CPF: " + i.cpf);  
        System.out.println("Data de nascimento: " + i.data_nascimento.toString());  
    }  
}
```

Deve-se saber que todos os métodos e atributos públicos e protegidos da superclasse (classe principal) serão herdados, caso utilize-se o extends Classe na definição de uma nova superclasse. Logo, os métodos e atributos privados (private) não serão herdados, e não teremos acesso a eles nem com a utilização da palavra super. Além da classe derivada herdar os comportamentos públicos da classe base, nada (ou quase nada) impede que façamos de novo os métodos e atributos que quisermos na subclasse.