

**Instituto Federal**  
Campus Goiânia

**Bacharelado em Sistemas de Informação**

**POO I**

**Prof. Dory Gonzaga Rodrigues**



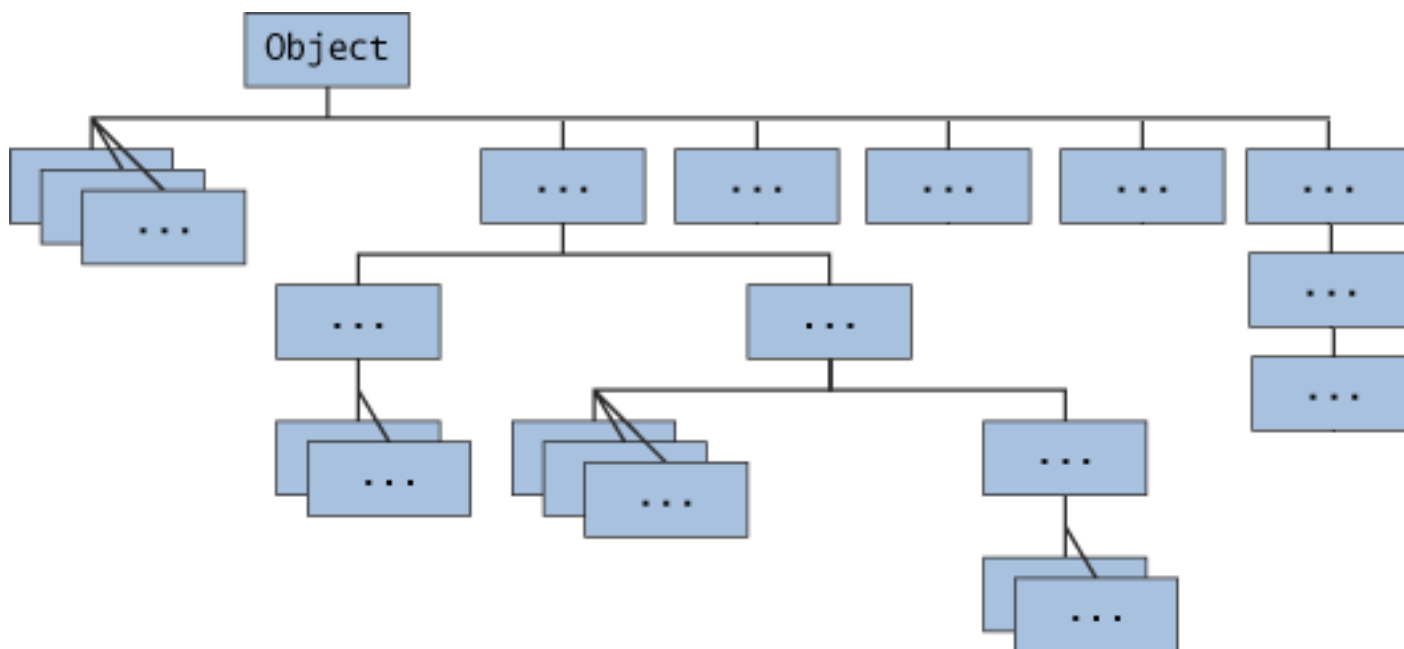
## Ementa

- O pacote java.lang e
- Classe Object

**Ementa** **Pacote java.lang**

## O pacote java.lang

O pacote java.lang contém todas as classes da base da linguagem do Java, ou seja, as classes que pertencem ao pacote java.lang representam a linguagem Java na sua essência.





Ementa

Pacote java.lang

Classe Object

## A Classe Object

A classe Object é a principal e mais básica classe do Java. Todas as outras a tem como origem e, portanto, **herdam** seus métodos.

Todos as classes herdam os métodos de Object.

**Object** é a raiz da hierarquia de classes do Java, a superclasse de todas as classes, direta ou indiretamente.

Sendo assim, Object define alguns comportamentos comuns que todos objetos devem ter, como a habilidade de serem comparados uns com os outros, utilizando o método **equals()**, a habilidade de serem representados como texto, através do método **toString()**, e possuírem um número que identifica suas posições em coleções baseadas em hash, com seu método **hashCode()**.



Ementa

Pacote java.lang

Classe Object

## Métodos da Classe Object

### toString()

O método `toString()` retorna um texto que representa o objeto da sua classe. Retorna um texto que contenha a informação que está contida e representa o objeto.



Ementa

Pacote java.lang

Classe Object

## Métodos da Classe Object

### `equals(Object obj)`

O método `equals()` define a identidade de um objeto durante a execução de uma aplicação. Este método indica se um objeto X é “igual” a um objeto Y.

Porém, esta igualdade é interpretada de forma específica para cada classe que implementa este método.

Por padrão, o método `equals()` de **Object** utiliza apenas o operador de igualdade `==` nas comparações. Portanto, dois objetos são considerados iguais apenas se as duas referências (ponteiros) apontam para o mesmo objeto.



Ementa

Pacote java.lang

Classe Object

## Métodos da Classe Object

### `equals(Object obj)`

Por vários motivos, você normalmente buscará a **igualdade “lógica”**, onde duas instâncias sejam consideradas “iguais” quando o valor das propriedades sejam o mesmo.

Nestes casos, devemos **sobrescrever o método `equals()`**. Para evitar que ocorra a sobrecarga ao invés da sobrescrita, sempre utilize a anotação **`@Override`**. Ela garante, em tempo de compilação, que o método de fato esteja sendo sobrescrito.



Ementa

Pacote java.lang

Classe Object

## Métodos da Classe Object

### hashCode()

O método `hashCode()` é utilizado para melhorar o desempenho em grandes coleções de dados.

Este método ajuda a organizar e recuperar os objetos de dentro das coleções.

A ideia é simples, criar um número que represente o objeto que será armazenado na coleção, facilitando assim o armazenamento, a localização e recuperação deste.





Ementa

Pacote java.lang

Classe Object

## Métodos da Classe Object

### hashCode()

De acordo com a especificação oficial, se dois objetos são considerados iguais de acordo com a implementação do método `equals(Object)`, então invocar o método `hashCode()` em cada um dos objetos deve produzir o mesmo número inteiro.

Além disso, como pode ser visto no exemplo, não é requerido/obrigatório que dois objetos considerados diferentes de acordo com a implementação do método `equals(Object)` devam retornar números inteiros distintos ao invocar `hashCode()`.

Veja no exemplo, onde dois objetos `String` contendo valores distintos “Paulo” e “Isabelle”, considerados diferentes pelo método `equals()`, porém obteve o mesmo número inteiro pelo `hashCode()`.

## Ementa » Pacote java.lang » Classe Object

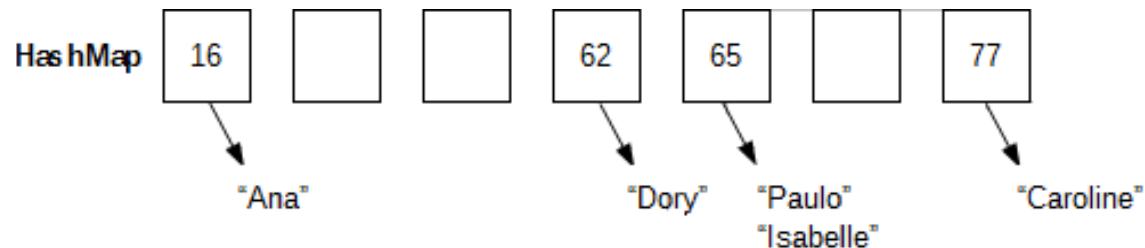
### Métodos da Classe Object

#### hashCode()

Coleção: Alunos

Nome	Algoritmo	HashCode
Dory	$D(4) + O(15) + R(18) + Y(25)$	62
Caroline	$C(3) + A(1) + R(18) + O(15) + L(12) + I(9) + N(14) + E(5)$	77
Paulo	$P(16) + A(1) + U(21) + L(12) + O(15)$	65
Ana	$A(1) + N(14) + A(1)$	16
Isabelle	$I(9) + S(19) + A(1) + B(2) + E(5) + L(12) + L(12) + E(5)$	65

Considerando:  $A = 1, B = 2 \dots Y = 25, Z = 26$





Ementa

Pacote java.lang

Classe Object

## Métodos da Classe Object

### **clone()**

O método clone() cria e retorna uma cópia desse objeto.

### **finalize()**

O método finalize() é utilizado pelo coletor de lixo determinando que não existe mais referências ao objeto. Libera a memória do objeto.

### **getClass()**

O método getClass() retorna a classe do objeto em tempo de execução.



Ementa

Pacote java.lang

Classe Object

## Métodos da Classe Object

### **notify()**

O método `notify()` acorda uma thread que está esperando o controle de um objeto.

### **notifyAll()**

O método `notifyAll()` acorda TODAS as threads que estão esperando o controle de um determinado objeto.

### **wait()**

O método `wait()` faz com que a thread libere o objeto e espere (durma) até que outra thread, através do mesmo objeto, a notifique que ela já pode voltar a executar (acordar).



Ementa

Pacote java.lang

Classe Object

## Exemplo do uso da Classe Object

```
public class ClasseObject {  
  
    public static void main (String[] args) {  
  
        Object o1 = new Object();  
        Object o2 = new Object();  
  
        System.out.println(o1.getClass());  
        System.out.println(o1.toString());  
        System.out.println(o1.hashCode());  
        System.out.println(o1.equals(o2));  
    }  
}
```

**Saída - Pacote.java.lang (run) X**

run:

class java.lang.Object

java.lang.Object@15db9742

366712642

false