



**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
São Paulo

Linguagem de Programação 2

Prof. Dr. Domingos Bernardo Gomes Santos

Ementa

- Principais conceitos sobre a classe **Object**;
- Métodos de **Object**;
- Escrevendo o método **toString**;
- Herança entre classes Java;
- Escrevendo o método **equals**;
- Exercícios.

Conceitos da classe Object

- Toda e qualquer classe elaborada através de códigos Java é subclasse da **Object**;
- Quando não utilizamos explicitamente o comando **extends Object**, a máquina virtual Java inclui automaticamente a herança da classe **Object**;
- A classe **Object** é a classe root de todas classes Java.

Métodos de Object

- **getClass()** este método retorna informações do objeto atual, como o nome da classe;
- **clone()** retorna uma referência - ou cópia - de um objeto;
- **toString()** retorna uma string que representa o objeto;
- **equals(Object obj)** faz a comparação entre dois Objects, e retorna true para objetos iguais e false para objetos diferentes;
- **hashCode()** retorna um inteiro único de cada objeto, muito usado pelas classes Collections.

Herança explícita de Object

```
1  package aula0;  
2  
3  public class Carro extends Object {  
4  
5      public String modelo;  
6  
7  public Carro(){  
8  
9      this.modelo = "";  
10  
11  }  
12  
13  }  
14
```

Escrevendo o toString

```
1 package aula1;
2
3 public class Veiculo {
4
5     private String modelo;
6
7     public Veiculo() {
8
9         this.modelo = "";
10    }
11
12    public String getModelo() {
13        return modelo;
14    }
15    public void setModelo(String modelo) {
16        this.modelo = modelo;
17    }
18
19    @Override
20    public String toString() {
21        return "[modelo=" + modelo + "]";
22    }
23
24 }
```

Herança entre classes

```
1 package aula1;
2
3 public class Carro extends Veiculo {
4
5     private String placa;
6
7     public Carro() {
8
9         super();
10
11         this.placa = "";
12     }
13
14     public String getPlaca() {
15         return placa;
16     }
17
18     public void setPlaca(String placa) {
19         this.placa = placa;
20     }
21
22     @Override
23     public String toString() {
24         return this.getClass() + super.toString() + "[placa=" + placa + "]";
25     }
26
27 }
28
```

Testando métodos de Object

```
1 package aula1;
2
3 public class TesteObject {
4
5     public static void main(String[] args) {
6
7         Veiculo v1 = new Veiculo();
8
9         Carro c1 = new Carro();
10        Carro c2 = new Carro();
11
12        System.out.println("\nv1.getClass(): " + v1.getClass());
13        System.out.println("\nc1.getClass(): " + c1.getClass());
14        System.out.println("\nc2.getClass(): " + c2.getClass());
15
16        System.out.println("\nv1.toString: " + v1.toString());
17        System.out.println("\nc1.toString: " + c1.toString());
18        System.out.println("\nc2.toString: " + c2.toString());
19
20        System.out.println("\nc1.equals(c2): " + c1.equals(c2));
21        System.out.println("\nc2.equals(c2): " + c2.equals(c1));
22
23        System.out.println("\nv1.hashCode(): "+ v1.hashCode());
24        System.out.println("\nc1.hashCode(): "+ c1.hashCode());
25        System.out.println("\nc2.hashCode(): "+ c2.hashCode());
26
27    }
28
29 }
```


Testando métodos de Object

Console

<terminated> TesteObject [Java Application] /opt/java/jdk1.8.0_121/bin/java (7 de fev de 2017 15:36:48)

v1.getClass(): class aula1.Veiculo

c1.getClass(): class aula1.Carro

c2.getClass(): class aula1.Carro

v1.toString: [modelo=]

c1.toString: class aula1.Carro[modelo=][placa=]

c2.toString: class aula1.Carro[modelo=][placa=]

c1.equals(c2): false

c2.equals(c2): false

v1.hashCode(): 366712642

c1.hashCode(): 1829164700

c2.hashCode(): 2018699554

Escrevendo método equals()

```
1 package aula1;
2
3 public class Aviao {
4
5     private String fabricante;
6     private String modelo;
7
8     public Aviao( String fabricante, String modelo ) {
9         this.fabricante = fabricante;
10        this.modelo = modelo;
11    }
12
13    public String getFabricante() {
14        return fabricante;
15    }
16    public void setFabricante(String fabricante) {
17        this.fabricante = fabricante;
18    }
19    public String getModelo() {
20        return modelo;
21    }
22    public void setModelo(String modelo) {
23        this.modelo = modelo;
24    }
25
26    @Override
27    public boolean equals(Object obj) {
28
29        Aviao a1 = (Aviao) obj;
30
31        if( this.fabricante.equals( a1.getFabricante() ) && this.modelo.equals( a1.getModelo() ) )
32            return true;
33        else
34            return false;
35    }
36
37 }
```

Testando o método equals

```
1 package aula1;
2
3 public class TesteAviao {
4
5     public static void main(String[] args) {
6
7         Aviao a1 = new Aviao("airbus","a330");
8         Aviao a2 = new Aviao("airbus","a330");
9         Aviao a3 = new Aviao("boeing","777-300");
10
11         System.out.println("a1.equals(a2): " + a1.equals(a2) );
12         System.out.println("a2.equals(a1): " + a2.equals(a1) );
13         System.out.println("a3.equals(a1): " + a3.equals(a1) );
14
15     }
16
17 }
18
```

Console

```
<terminated> TesteAviao [Java Application] /opt/java/jdk1.8.0_121/bin/java (7 de fev de 2017 15:53:17)
a1.equals(a2): true
a2.equals(a1): true
a3.equals(a1): false
```

Exercícios