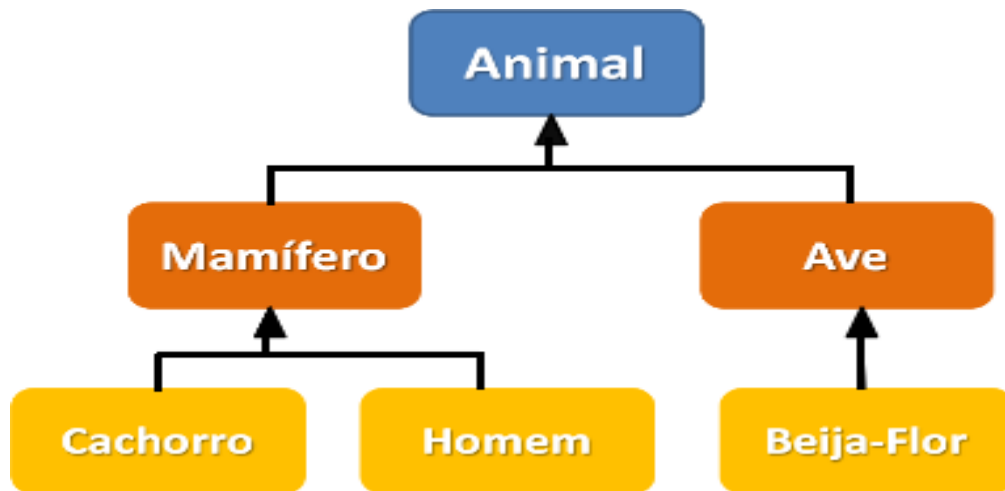


Dia 18 – Herança – Java

Vamos aprender na pratica todo conceito de Herança, É uma, Extend.



Conteúdo Herança Dia 18:

Como nos aprendemos vamos criar duas classes chamado Aluno e Professor, com os get,set e seus private com variável

Exemplo

Aluno	
nome	String
endereço	String
telefone	String
cpf	String
cursos	String[]
notas	String[]
calcularMedia()	double
verificarAprovado()	double

Professor	
nome	String
endereço	String
telefone	String
cpf	String
departamento	String
nomeCurso	String
salario	double
calcularSalarioLiquido()	double

Dia 18 – Herança – Java

```
1. public class Aluno{
2.     private String nome;
3.     private String endereco;
4.     private String telefone;
5.     private String cpf;
6.     private String curso;
7.     private double[] notas;
8.
9.     public String getNome() {
10.         return nome;
11.     }
12.     public void setNome(String nome) {
13.         this.nome = nome;
14.     }
15.     public String getEndereco () {
16.         return endereco;
17.     }
18.     public void setEndereco(String endereco) {
19.         this.endereco = endereco;
20.     }
21.
22.     public String getTelefone() {
23.         return telefone;
24.     }
25.     public void setTelefone(String telefone) {
26.         this.telefone = telefone;
27.     }
28.     public String getCpf() {
29.         return cpf;
30.     }
31.     public void setCpf(String cpf) {
32.         this.cpf = cpf;
33.     }
34.
35.     public double[] getNotas() {
36.         return notas;
37.     }
38.     public void setNotas(double[] notas) {
39.         this.notas = notas;
40.     }
41.     public double calcularMedia(){
42.         return 0;
43.     }
44.     public boolean verificarAprovado(){
45.         return true;
46.     }
47. }
```

Agora a classe professor.

```
1. public class Professor{
2.     private String nome;
3.     private String endereco;
4.     private String telefone;
5.     private String cpf;
6.     private double salario;
7.     private String nomeCurso;
8.
9.     public String getNome() {
10.         return nome;
11.     }
```

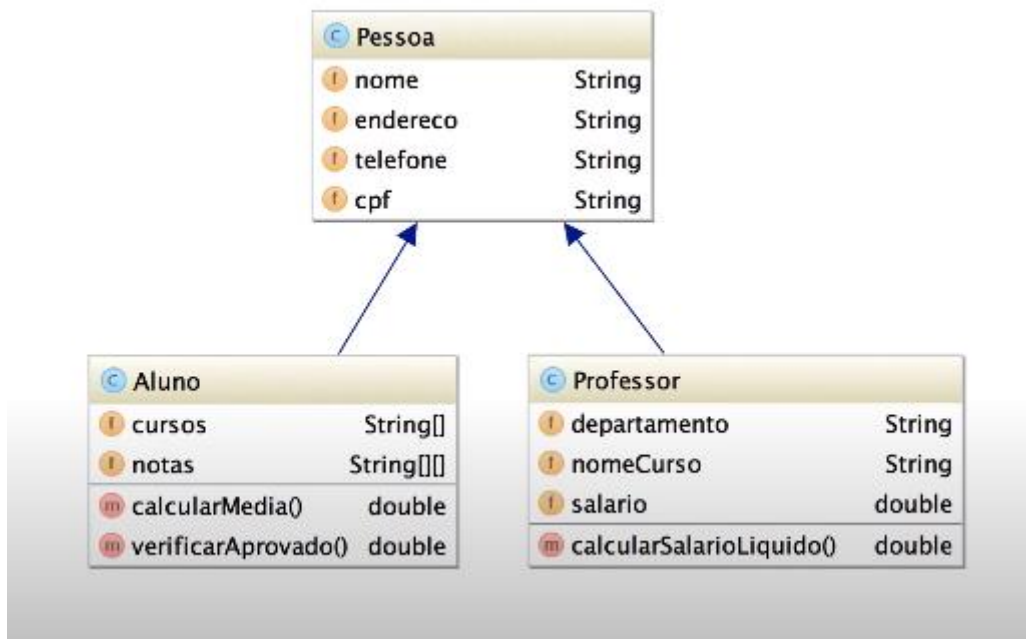
Dia 18 – Herança – Java

```
11. }
12. public void setNome(String nome) {
13.     this.nome = nome;
14. }
15. public String getEndereco () {
16.     return endereco;
17. }
18. public void setEndereco(String endereco) {
19.     this.endereco = endereco;
20. }
21.
22.     public String getTelefone() {
23.     return telefone;
24. }
25. public void setTelefone(String telefone) {
26.     this.telefone = telefone;
27. }
28.     public String getCpf() {
29.     return cpf;
30. }
31. public void setCpf(String cpf) {
32.     this.cpf = cpf;
33. }
34.
35. public double[] getNotas() {
36.     return notas;
37. }
38. public void setNotas(double[] notas) {
39.     this.notas = notas;
40. }
41.
42. }
```

Podemos observar que em Aluno e Professor tem a mesma variável nome, endereço, telefone e cpf. Pois todas as pessoas tem esses atributo, ou seja se tiver que criar uma classe chamada Diretora, teria que colocar o nome, endereço, telefone e cpf igual.

Para solucionar isso vamos usar a Herança que todas as pessoas vão herdar esses atributos.

Dia 18 – Herança – Java



Isso é ótimo porque não precisa ficar duplicando variável, acabando ter problema em armazenar quantidade extensa no servidor. E para utilizar a herança você irá utilizar o “extends” que herda todos os atributo dessa classe menos os private, somente o public,protected.

```
public class Professor extends Pessoa {

    private double salario;
    private String nomeCurso;

    public double getSalario() {
        return salario;
    }
    public void setSalario(double salario) {
        this.salario = salario;
    }
    public String getNomeCurso() {
        return nomeCurso;
    }
    public void setNomeCurso(String nomeCurso) {
        this.nomeCurso = nomeCurso;
    }

    public double calcularSalarioLiquido() {
        return 0;
    }

}

public class Pessoa {

    private String nome;
    private String endereco;
    private String telefone;
    private String cpf;
```

Dia 18 – Herança – Java

```
private String telefoneCelular;

public String getNome() {
    return nome;
}
public void setNome(String nome) {
    this.nome = nome;
}
public String getEndereco() {
    return endereco;
}
public void setEndereco(String endereco) {
    this.endereco = endereco;
}
public String getTelefone() {
    return telefone;
}
public void setTelefone(String telefone) {
    this.telefone = telefone;
}
public String getCpf() {
    return cpf;
}
public void setCpf(String cpf) {
    this.cpf = cpf;
}
public String getTelefoneCelular() {
    return telefoneCelular;
}
public void setTelefoneCelular(String telefoneCelular) {
    this.telefoneCelular = telefoneCelular;
}

}

public class Aluno extends Pessoa {

    private String curso;
    private double[] notas;

    public String getCurso() {
        return curso;
    }
    public void setCurso(String curso) {
        this.curso = curso;
    }
    public double[] getNotas() {
        return notas;
    }
    public void setNotas(double[] notas) {
        this.notas = notas;
    }

    public double calcularMedia(){
        return 0;
    }

    public boolean verificarAprovado(){
        return true;
    }
}
```

Dia 18 – Herança – Java

```
}  
}
```

Agora podemos usar os métodos get e set em aluno chamando nome que esta herdado de Pessoa.

Ou seja Aluno é uma pessoa(recebe os atributo de pessoa) e Professor é uma pessoa(recebe os atributo de pessoa).

```
package com.loiane.cursojava.aula37;  
  
public class Teste {  
  
    public static void main(String[] args) {  
  
        /*Aluno aluno = new Aluno();  
  
        Professor professor = new Professor();  
  
        Aluno aluno = new Aluno();  
  
        aluno.getNome();  
  
    }  
}
```

Exercicio:

- 1) Tenho uma classe chamado Aluno como eu herdo de uma classe chamada Pessoa
A) **public class Aluno extends Pessoa{}**
B) public class Pessoa extends Aluno {}
C) public class Aluno implements Pessoa{}
D) public class Pessoa extends Aluno {}

- 2) Tenho uma classe chamado Professor como eu herdo de uma classe chamada Pessoa
A) **public class Professor extends Pessoa{}**
B) public class Pessoa extends Professor{}
C) public class Professor implements Pessoa{}
D) public class Pessoa extends Professor{}

Dia 18 – Herança – Java

- A) De acordo com a classe como seria de forma mais correta utilizando a herança

Exemplo

Aluno	
nome	String
endereço	String
telefone	String
cpf	String
cursos	String[]
notas	String[][]
calcularMedia()	double
verificarAprovado()	double

Professor	
nome	String
endereço	String
telefone	String
cpf	String
departamento	String
nomeCurso	String
salário	double
calcularSalarioLiquido()	double

- A) Está correto não é necessário nem uma modificação.
- B) Criar uma classe chamado Pessoa com as variáveis nome,endereço,telefone e cpf
- C) Criar uma classe chamado Pessoa com as variáveis nome,endereço,telefone e cpf enquanto o Professor e Aluno utiliza o extends, não podendo acessar os privates de Aluno e Professor**
- D) Criar uma classe chamado Pessoa com as variáveis nome,endereço,telefone e cpf enquanto o Professor e Aluno utiliza o extends, podendo acessar os privates de Aluno e Professor
-

Prova

1)

1. class Clidder {
2. private final void flipper() { System.out.println("Clidder"); }
3. }
4. public class Clidlet extends Clidder {
5. public final void flipper() { System.out.println("Clidlet"); }
6. public static void main(String [] args) {
7. new Clidlet().flipper();
8. }}

Dia 18 – Herança – Java

Qual é o resultado?

A. Clidlet

B. Clidder

C. Clidder

Clidlet

D. Clidlet

E. Compilador falha

2) Quais afirmações são verdadeiras? (Escolha todas as opções aplicáveis.)

A. Coesão é o princípio OO mais associado a ocultar detalhes de implementação

B. Coesão é o princípio OO mais associado a garantir que as classes saibam

sobre outras classes apenas por meio de suas APIs

C. Coesão é o princípio OO mais intimamente associado a garantir que uma classe seja

projetado com um propósito único e bem focado

D. Coesão é o princípio OO mais intimamente associado a permitir que um único objeto seja

visto como tendo muitos tipos

3)

Dado o seguinte,

1. classe X {void do1 () {}}

2. classe Y estende X {void do2 () {}}

3 -

4. classe Chrome {

1. public static void main (String [] args) {

2. X x1 = novo X ();

3. X x2 = novo Y ();

9. Y y1 = novo Y ();

10. // insira o código aqui

4. }

5. }

Dia 18 – Herança – Java

Que, inserido na linha 9, irá compilar? (Escolha todas as opções aplicáveis.)

A. `x2.do2 ();`

B. `(Y) x2.do2 ();`

C. `((Y) x2) .do2 ();`

D. Nenhuma das declarações acima irá compilar