



/Exemplo de Heranca

```
package aula7;
```

```
//Subclasse
```

```
public class Principal {
```

```
    public static void main(String[] args) {
```

```
        Principal principal =  
            new Principal();
```

```
        principal.iniciar();
```

```
    }
```

```
    public void iniciar(){
```

```
        /*Girino girino =
```

```
            new Girino("GIRINO",  
                "VERDE");
```

```
            System.out.println(girino.getNome());
```

```
        */
```

```
        CoelhoRajado coelhoRajado=
```

```
            new CoelhoRajado(  
                "COELHO RAJADO",  
                "RAJADO",  
                "MUITOS",  
                4
```

```
            );
```

```
        System.out.println(coelhoRajado );
```

```
    }
```

```
}
```

```
//-----
```

```
package aula7;
```

```
//Subclasse
public class CoelhoRajado
    extends Coelho{

    public CoelhoRajado(
        String nome,
        String cor,
        String pelos,
        int pernas){

super(nome,cor,pelos,pernas);

    }

    //Sobrescrita
    public String toString(){

        return this.getNome() + "\n" +
            this.getCor() + "\n" +
            this.pelos + "\n" +
            this.pernas;
    }

}
```

```
//-----

package aula7;

//Subclasse
public class Coelho
    extends Animal{

    protected String pelos;

    public Coelho(
        String nome,
        String cor,
        String pelos,
        int pernas){

        super(nome,cor,pernas);
        this.pelos=pelos;

        //Heranca
        //setCor(cor);
```

```
//Heranca
//setNome(nome);

}
}

//----
```

```
package aula7;
//Superclasse
public class Animal {

    private String nome;
    private String cor;
    protected int pernas;

    public Animal(
        String nome,
        String cor,
        int pernas
    ){
        //Heranca
        setCor(cor);
        //Heranca
        setNome(nome);

        this.pernas=pernas;
    }

    public String getNome(){
        return this.nome;
    }
    public String getCor(){
        return this.cor;
    }
    public void setCor(String cor){
        this.cor=cor;
    }
    public void setNome(String nome){
        this.nome=nome;
    }
}
```

- 1) (Online) Acesse o link: <https://codeboard.io/projects/343113>
  - a) Realize as atividades do link.

```

/Estudante
public class Estudante
    extends MembroUniversitario{
    private String nome;
    private int idade;
    private String CPF;

    public Estudante(String nome){
        this.nome=nome;
        this.idade=0;
        this.CPF="";
    }
    public Estudante(String nome, int idade){
        this.nome=nome;
        this.idade=idade;
        this.CPF="";
    }
    public Estudante(String nome, int idade, String CPF){
        this.nome=nome;
        this.idade=idade;
        this.CPF=CPF;
    }

    public String toString(){
        return this.nome + "\n" + this.idade + "\n" + this.CPF;
    }
}

```

```

/MembroUniversitario
public class MembroUniversitario{

}

```

```

//Principal
import java.util.ArrayList;
public class Principal {

    public static void main(String[] args) {
        Principal p = new Principal();
        p.iniciar();
    }
    public void iniciar(){

        ArrayList<Estudante> lista = new ArrayList<>();
        ArrayList <Tecnico> listaT = new ArrayList<>();
    }
}

```

```

Estudante joao = new Estudante("JOAO");
Estudante maria = new Estudante ("MARIA", 20);
Estudante jose = new Estudante ("JOSE", 20, "123.456.789-0");

lista.add(joao);
lista.add(maria);
lista.add(jose);
for(Estudante estudante : lista)
System.out.println(estudante);

Tecnico joaoT = new Tecnico();
Tecnico mariaT = new Tecnico();
Tecnico joseT = new Tecnico();

listaT.add (joaoT);
listaT.add (mariaT);
listaT.add (joseT);
for(Tecnico tecnico : listaT) System.out.println(tecnico);
}
}

```

```

//Tecnico
public class Tecnico
extends MembroUniversitario{

public String toString(){
return "TECNICO";
}
}

```

2) (Netbeans) Herança: Implemente o diagrama UML de classes da Figura 1:

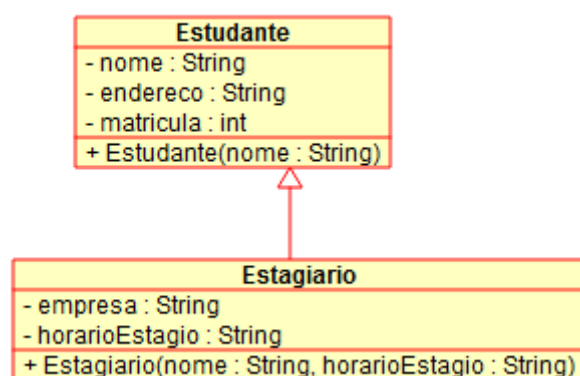


Figura 1 - Diagrama UML de Classes.

3) (NetBeans) Herança: Implemente o diagrama UML de classes da Figura 2:

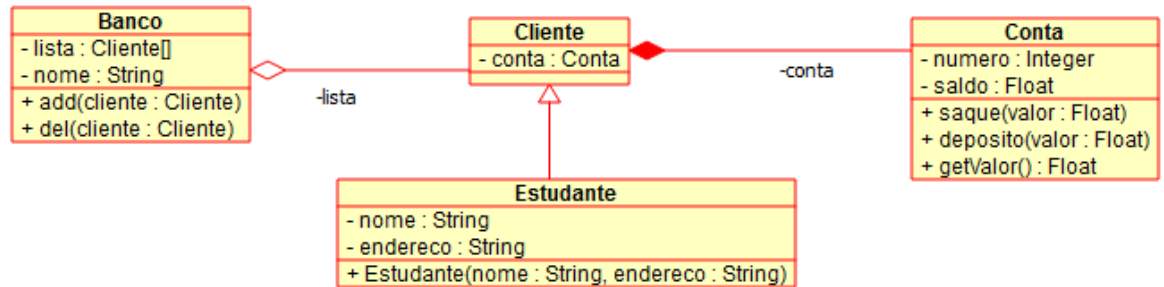


Figura 2 - Diagrama UML de Classes.

4) (NetBeans) Herança: Implemente o diagrama UML de classes da Figura 3:

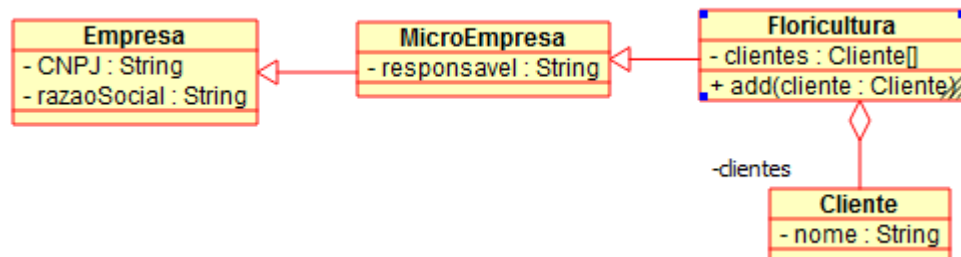


Figura 3 - Diagrama UML de Classes.

5) (NetBeans) A partir do enunciado, identifique:

	Uma transportadora é uma microempresa. A microempresa é uma empresa.
Classe	
Subclasse	
Superclasse	

6) (NetBeans) A partir do enunciado, identifique:

	Uma pilha é uma estrutura de dados. Uma estrutura de dados é uma Lista.
Classes	
Subclasse	
Superclasse	