



Compartilhar o seu link com: [luciorocha @ professores.utfpr.edu.br](mailto:luciorocha@professores.utfpr.edu.br)

Alexandre Calisto: [Aula 8](#)

Ruan Mateus Trizotti:

MARIA EDUARDA PEDROSO :

Arnald Souza: [Aula 8 - 06/09/2022](#)

Gustavo Naoki Jodai Kurozawa: [Aula 8 - 06/09/2022 - Exercícios](#)

Lucas Santana: [Aula 8 - Lucas Santana](#)

Guilherme Ramalho: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Raphael Hideyuki Uematsu: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Mariana Pedroso Naves: [Aula 8 - POO](#)

Matheus Henrique De A. Pereira: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Gabriel Candelária: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Camila Costa: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Manuela Torres Vieira: [Aula 8 - POO](#)

Michael Pariz Pereira:

https://docs.google.com/document/d/1OA4i_hm6y_8QiWNZ2bWo0x5DZNzsT9UM49gSHOU/DsOg/edit?usp=sharing

Leonardo G. Fagote [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Rafael Zaupa Watanabe: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Iago Macarini Brito: [Aula 8 - POO](#)

Ruan Perondi Urbanjos: [Copy of POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Vitor Luis de Queiroz Batista: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Julio Farias: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Rodrigo Leandro Benedito: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Felipe Antonio Magro: [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

Erik Noda: [Aula8](#)

João Pedro de Paula : [Cópia de POCO4A - Aula 8 - 06/09/2022 - Exercícios](#)

(...Continuação dos exercícios da aula anterior)

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

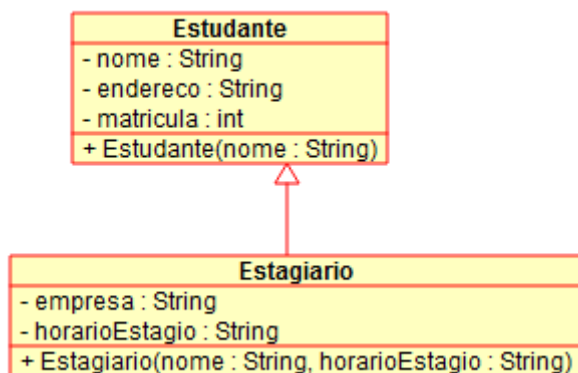


Figura 1 - Diagrama UML de Classes.

```

//Estagiario → Estudante (Generalização)
//Estudante → Estagiário (Especialização)

package aula8;
//SUBCLASSE
public class Estagiario extends Estudante{
    private String empresa;
    private String horarioEstagio;

    public Estagiario(String empresa,String horarioEstagio){
        this.empresa = empresa;
        this.horarioEstagio = horarioEstagio;
    }

    public Estagiario(String nome, String empresa, String horarioEstagio) {
        super(nome);//construtor passando como argumento
        this.empresa = empresa;
        this.horarioEstagio = horarioEstagio;
    }

    public Estagiario(String nome, String empresa, String horarioEstagio,float salario)
    {
        super(nome,salario);//construtor passando como argumento
    }
  
```

```

    this.empresa = empresa;
    this.horarioEstagio = horarioEstagio;
}
}

```

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
 */

```

```
package aula8;
```

```
public class Principal {
```

```
    public Principal(){
```

```
        Estagiario jorge = new Estagiario("caralh", "manha");
        jorge.setNome("madu");
```

```
        Estagiario maria = new Estagiario("mary","caralh", "manha");
```

```
        Estagiario ruan = new Estagiario("ruan","caralh", "manha",2800);
```

```
    }
```

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

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

```
    }
```

```
    private void iniciar() {
```

```
    }
```

```
}
```

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license

```

```

 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this
template

```

```
 */
```

```
package aula8;
```

```
/**
```

```
 * SUPERCLASSE
```

```
 */
```

```

public class Estudante {
    private String nome;
    private String endereco;
    private int matricula;
    private float salario;

    public Estudante(String nome){
        this.nome = nome;
    }

    public Estudante(String nome,float salario){
        this.nome = nome;
        this.salario = salario;
    }

    public void setNome(String nome){
        this.nome = nome;
    }
}

```

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

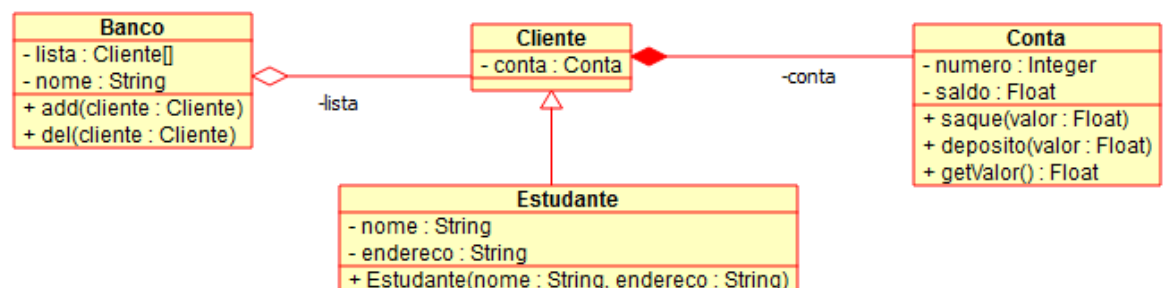


Figura 2 - Diagrama UML de Classes.

```

public class Banco {
    private ArrayList<Cliente> lista;
    private String nome;

    public Banco(){
        lista = new ArrayList<>();
    }
}

```

```

        this.nome = "BANCO";
    }
    public void add(Cliente cliente){
        lista.add( cliente );
    }
    public void del(Cliente cliente) {
    }
} //fim classe

```

```

//Subclasse
public class Cliente {
    private Conta conta;    //Composicao

    public Cliente( Conta conta ){
        this.conta = conta;
    }
} //fim classe

```

```

public class Estudante extends Cliente {
    private String nome;
    private String endereco;
    public Estudante( String nome, String endereco ){
        this.nome = nome;
        this.endereco = endereco;
    }
}

```

```

public class Conta {
    private int numero;
    private float saldo;

    public Conta( int numero, float saldo ){
        this.numero = numero;
        this.saldo = saldo;
    }

    public void saque(float valor){
        this.saldo -= valor;
    }
    public void deposito(float valor){
        this.saldo += valor;
    }
    public float getValor(){
        return this.saldo;
    }
}

```

```

    }
} //fim classe

```

```

public class Banco {
    private ArrayList<Cliente> lista;
    private String nome;

    public Banco(){
        lista = new ArrayList<>();
        this.nome = "BANCO";
    }
    public void add(Cliente cliente){
        lista.add( cliente );
    }
    public void del(Cliente cliente) {
    }
} //fim classe

//Subclasse
public class Cliente {
    private Conta conta; //Composicao

    public Cliente( Conta conta ){
        this.conta = conta;
    }
    public Conta getConta(){
        return this.conta;
    }
}

} //fim classe

```

```

public class Estudante extends Cliente {
    private String nome;
    private String endereco;

```

```

        public Estudante( String nome, String endereco ){
            this.nome = nome;
            this.endereco = endereco;
        }
        public Estudante( String nome, String endereco, Conta conta ){
            super( conta );
            this.nome = nome;
            this.endereco = endereco;
        }
    }

public class Conta {
    private int numero;
    private float saldo;

    public Conta( int numero, float saldo ){
        this.numero = numero;
        this.saldo = saldo;
    }

    public void saque(float valor){
        this.saldo -= valor;
    }
    public void deposito(float valor){
        this.saldo += valor;
    }
    public float getValor(){
        return this.saldo;
    }
}

//fim classe

//—
public class Principal {

    public Principal(){

        Conta conta = new Conta(123, 10000f);
        Estudante joao = new Estudante( "JOAO", "Rua 123", conta );
        joao.getConta().deposito(50000f);

    }
}

```

```

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

```

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

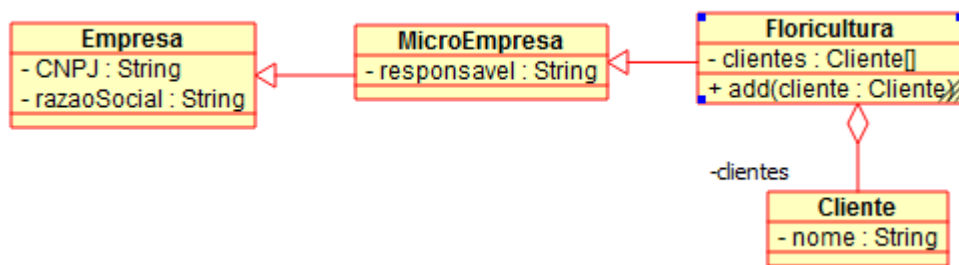


Figura 3 - Diagrama UML de Classes.

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

	Uma transportadora é uma microempresa. A microempresa é uma empresa.
Classe	Empresa transportadora microempresa
Subclasse	transportadora (microempresa superclasse) microempresa (empresa superclasse) transportadora (empresa superclasse)
Superclasse	microempresa (transportadora) empresa (microempresa transportadora)

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

	Uma pilha é uma estrutura de dados. Uma estrutura de dados é uma Lista.
Classes	lista, pilha, lista
Subclasse	pilha (em relação a estrutura de dados)

	estrutura de dados (em relação a lista) pilha (em relação a lista indiretamente)
Superclasse	estrutura de dados(em relação a pilha) lista(em relação a estrutura de dados e a pilha)