



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

Alexandre Calisto: [Aula 6](#)

Matheus Henrique A. Pereira: [Cópia de POCO4A - Aula 6 - 30/08/2022 - Exercícios](#)

Gustavo Naoki Jodai Kurozawa: [Cópia de POCO4A - Aula 6](#)

Plinio Koiana: [Cópia de POCO4A - Aula 6 - 30/08/2022 - Exercícios](#)

Arnald Souza: [Aula 6 - 30/08/2022](#)

Felipe Antonio Magro: [Cópia de POCO4A - Aula 6 - 30/08/2022 - Exercícios](#)

Lucas Santana: [aula 6 - Lucas Santana](#)

Lucas Prado - Erik Noda: [Cópia de POCO4A - Aula 6 - 30/08/2022 - Exercícios](#)

Rodrigo Leandro Benedito: [Cópia de POCO4A - Aula 6 - 30/08/2022 - Exercícios](#)

Leonardo G. Fagote [Cópia de POCO4A - Aula 6 - 30/08/2022 - Exercícios](#)

1) (Online) Acesse o link: <https://codeboard.io/projects/343077>

a) Realize as atividades do link.

```
public class Principal {  
  
    public static void main(String[] args) {  
        Estudante ruan = new Estudante("Ruan");  
        Estudante isa = new Estudante("isa",22);  
        Estudante natan = new Estudante("natan",21,"46393399956");  
  
    }  
}
```

```
public class Estudante {  
    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 int imprimir(){
        return this.idade;
    }

    public String imprimir(String tudo){
        return this.nome + " " + this.idade + " " + this.CPF;
    }

    public String imprimir(int segundo){
        return this.nome + " " + this.idade;
    }

    public int getIdade(){
        return this.idade;
    }
}

```

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



Figura 1 - Diagrama UML de Classes.

```

package aula6;

import java.util.ArrayList;

public class Curso {
    private String nome;
    private ArrayList<Estudante> lista; //declaracao do objeto
}

```

```

public Curso(String nome){
    this.nome = nome;
    lista = new ArrayList<Estudante>(); //alocacao dinamica de memoria
}
public void add(Estudante estudante){
    this.lista.add( estudante );
}
public ArrayList<Estudante> getEstudantes(){
    return this.lista;
}
}

```

```

//---
package aula6;
public class Estudante {
    private String nome;

    public Estudante(String nome){
        this.nome = nome;
    }
    public String toString(){
        return "Nome: " + this.nome;
    }
}

```

```

//-----

import aula6.Curso;
import aula6.Estudante;
import java.util.ArrayList;
public class Principal {

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

    public void iniciar(){

        Estudante joao = new Estudante("JOAO");
        Curso poo = new Curso("POO");
        poo.add( joao ); //objeto foi adicionado ah lista do Curso

        ArrayList<Estudante> conteudoLista = poo.getEstudantes();
        for ( Estudante estudante : conteudoLista)

```

```

        System.out.println( estudante );
    }
}

```

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

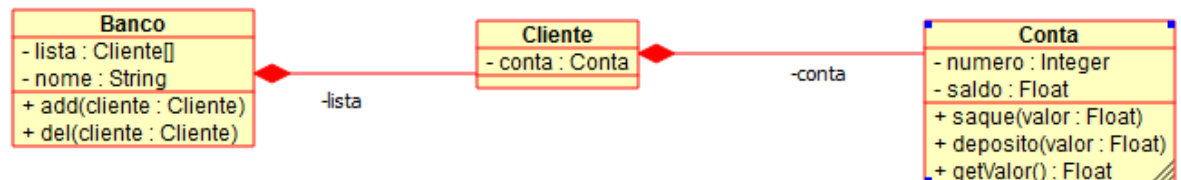


Figura 2 - Diagrama UML de Classes.

```

package aula6;

import java.util.ArrayList;

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

    public Banco(String nome){
        this.nome = nome;
        this.lista = new ArrayList<Cliente>();
    }
    public void add(Cliente cliente){
        this.lista.add( cliente );
    }
    public void del(int numero){

        for( Cliente cliente : lista )
            if ( cliente.getConta().getNumero() == numero )
                this.lista.remove(cliente );
    }
    public ArrayList<Cliente> getCientes(){
        return this.lista;
    }
}
//-----
package aula6;
public class Conta {

    private Integer numero;
    private Float saldo;

```

```

    public Conta ( Integer 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;
    }

    public int getNumero(){
        return this.numero;
    }

    //Float: é uma classe (wrapper / encapsuladora de tipo )
    //Exemplo: Float saldo;
    //          saldo.parseFloat("1.23");
    //float: é um tipo primitivo de dados.
}

//-----
package aula6;
public class Cliente {
    private Conta conta;
    private String nome;

    public Cliente( String nome, Conta conta ) {
        this.nome = nome;
        this.conta = conta;
    }

    public String toString(){
        return "Nome: " + this.nome;
    }

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

```

```

}

//-----
package aula6;

import java.util.ArrayList;

public class Principal {

    public Principal(){
        Conta conta = new Conta( 123, 100.0f );
        Cliente joao = new Cliente( "JOAO", conta );
        Cliente maria = new Cliente( "MARIA", conta );
        Banco cc = new Banco( "CC" );

        cc.add( joao );
        cc.add( maria );
        //A partir do Banco, quero imprimir o nome do Cliente
        ArrayList<Cliente> lista = cc.getClientes();
        for( Cliente cliente : lista )
            System.out.println( cliente );

        /*cc.del(123);
        try {
            for( Cliente cliente : lista )
                System.out.println( cliente );
        } catch(Exception e) {
            e.printStackTrace();
        }*/
    }

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

```

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

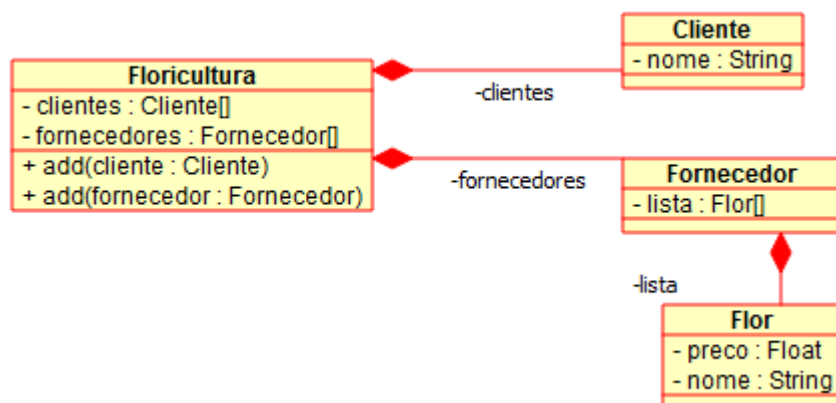


Figura 3 - Diagrama UML de Classes.

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

	Uma transportadora possui veículos de dois tipos: van e caminhão. A transportadora faz entregas de dois tipos de encomendas: normal e expresso.
Classes	
Atributos	
Comportamentos	

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

	Uma estrutura de dados do tipo pilha possui uma lista de dados do tipo inteiro. A pilha é do tipo FILO (First-In, Last-Out). As operações da pilha são: inserir (push) e remover (pop).
Classes	
Atributos	
Comportamentos	