

Mão na massa!

Faça uso de composição para resolver o problema de herança abaixo:

```
public class ProcessadorDeInvestimentos {

    public static void main(String[] args) {

        for (ContaComum conta : contasDoBanco()) {
            conta.rende();

            System.out.println("Novo Saldo:");
            System.out.println(conta.getSaldo());
        }
    }

    private static List<ContaComum> contasDoBanco() {
        return Arrays.asList(umaContaCom(100), umaContaCom(150), contaDeEstudanteCom(200));
    }

    private static ContaComum contaDeEstudanteCom(double amount) {
        ContaDeEstudante c = new ContaDeEstudante();
        c.deposita(amount);
        return c;
    }

    private static ContaComum umaContaCom(double valor) {
        ContaComum c = new ContaComum();
        c.deposita(valor);
        return c;
    }
}

public class ContaDeEstudante extends ContaComum {

    private int milhas;

    public void deposita(double valor) {
        super.deposita(valor);
        this.milhas += (int)valor;
    }

    public int getMilhas() {
        return milhas;
    }

    public void rende() {
        throw new RuntimeException("Não pode render");
    }
}

public class ContaComum {
```

```
protected double saldo;

public ContaComum() {
    this.saldo = 0;
}

public void deposita(double valor) {
    this.saldo += valor;
}

public void saca(double valor) {
    if (valor <= this.saldo) {
        this.saldo -= valor;
    } else {
        throw new IllegalArgumentException();
    }
}

public void rende() {
    this.saldo += this.saldo * 0.01;
}

public double getSaldo() {
    return saldo;
}
}
```

Responda

INSERIR CÓDIGO	FORMATAÇÃO
...	