



06

Mão na massa!

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

```
public class ProcessadorDeInvestimentos {

    static void Main(string[] args)
    {
        IList<ContaComum> contas = ContasDoBanco();

        foreach (ContaComum conta in contas)
        {
            conta.somaInvestimento();

            Console.WriteLine("Novo saldo: " + conta.Saldo );
        }

        Console.ReadLine();
    }

    private static IList<ContaComum> ContasDoBanco() {
        List<ContaComum> contas = new List<ContaComum>();
        contas.Add(umaContaComum(100));
        contas.Add(umaContaComum(150));
        contas.Add(umaContaEstudante(100));
        return contas;
    }

    private static ContaEstudante umaContaEstudante(double saldo)
    {
        ContaEstudante conta = new ContaEstudante();
        conta.Deposita(saldo);
        return conta;
    }

    private static ContaComum umaContaComum(double saldo)
    {
        ContaComum conta = new ContaComum();
        conta.Deposita(saldo);
        return conta;
    }
}

class ContaEstudante : ContaComum
{
    public int Milhas { get; private set; }

    public override void Deposita(double valor)
    {
        base.Deposita(valor);
        this.Milhas += (int)valor;
        throw new ArgumentException();
    }
}
```

```
    }

    class ContaComum
    {
        public double Saldo { get; private set; }

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

        public virtual void Deposita(double valor)
        {
            this.Saldo += valor;
        }

        public void Saca(double valor)
        {
            if (valor <= this.Saldo)
            {
                this.Saldo -= valor;
            }
            else
            {
                throw new ArgumentException();
            }
        }

        public void somaInvestimento()
        {
            this.Saldo += this.Saldo * 0.01;
        }
    }
}
```

Responda

INSERIR CÓDIGO	FORMATAÇÃO

