

Escrever código duplicado é algo que todo nós fazemos com frequência. Mas qual o problema disso? O problema é que sempre que precisarmos mexer em um lugar, precisaremos mexer no outro. Veja, por exemplo, as duas classes abaixo, que lidam com contas bancárias de Pessoa Física e Pessoa

Jurídica:

```
public class ContaPJ {

    private double saldo;
    private String titular;

    public ContaPJ(String titular, double saldoInicial) {
        this.titular = titular;
        saldo = saldoInicial;
    }

    public void saca(double valor) {
        saldo -= valor + 0.2;
    }

    public void deposita(double valor) {
        saldo += valor - 0.2;
    }

    public double getSaldo() {
        return saldo;
    }

    public String getTitular() {
        return titular;
    }
}

public class ContaPF {

    private double saldo;
    private String titular;

    public ContaPF(String titular, double saldoInicial) {
        this.titular = titular;
        saldo = saldoInicial;
    }

    public void saca(double valor) {
        saldo -= valor + 0.1;
    }

    public void deposita(double valor) {
        saldo += valor - 0.1;
    }

    public double getSaldo() {
        return saldo;
    }

    public String getTitular() {
        return titular;
    }
}
```

Veja que temos muito código repetido em ambas as classes. A começar pelos atributos: saldo e titular. Depois pela implementação dos métodos saca() e deposita() que são bem parecidos.

Precisamos fazer com que ambas as classes reutilizem código.

Uma maneira para tal é fazer uso de herança. Se criarmos uma classe abstrata pai e colocarmos o comportamento em ambos, veja como fica muito mais fácil mudar:

```
public abstract class ContaBancaria {
    protected double saldo;
    protected String titular;

    public ContaBancaria(String titular, double saldoInicial) {
        this.titular = titular;
        saldo = saldoInicial;
    }

    public void saca(double valor) {
        saldo -= valor;
    }

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

    public double getSaldo() {
        return saldo;
    }

    public String getTitular() {
        return titular;
    }
}

public class ContaPJ extends ContaBancaria {

    public ContaPJ(String titular, double saldoInicial) {
        super(titular, saldoInicial);
    }

    public void saca(double valor) {
        super.saca(valor + 0.2);
    }

    public void deposita(double valor) {
        super.deposita(valor - 0.2);
    }
}

public class ContaPF extends ContaBancaria {

    public ContaPF(String titular, double saldoInicial) {
        super(titular, saldoInicial);
    }

    public void saca(double valor) {
        super.saca(valor + 0.1);
    }

    public void deposita(double valor) {
        super.deposita(valor - 0.1);
    }
}
```

Veja que agora o código está escondido na classe pai, e as classes filhos reutilizam. Herança é uma maneira, mas não a única. Sempre que encontrar código duplicado entre duas classes, refatore. Para tal, use herança, composição, crie abstrações, faça uso de padrões de projeto. Mas não deixe código repetido!