

Benefícios da coesão

Preste atenção nos métodos da classe `BalancoEmpresa`. Note que cada um deles recebe uma quantidade razoável de parâmetros. O método `void pagaDivida(String cnpjCredor, double valor, String nomePagador, String cnpjPagador)`, por exemplo, recebe como argumentos o CNPJ do credor e os dados do pagamento a ser efetuado.

A maneira como está definida a assinatura do método pode gerar confusão no momento de seu uso. Imagine uma chamada do método `void pagaDivida(...)` como a seguinte.

```
...
new BalancoEmpresa().pagaDivida("00.000.000/0001-01", 500.0, "Empresa Pagadora", "00.000.000/0002-02", "Empre:");
...
```

Não fica imediatamente claro, ao nos depararmos com a lista de argumentos, qual é, por exemplo o CNPJ do credor e qual é o do pagador. Além disso, numa chamada do método onde os parâmetros aparecem em ordem equivocada, não contaremos com a ajuda do compilador para nos alertar sobre este erro. Por exemplo, ao trocarmos a ordem dos argumentos que representam o nome da empresa pagadora e seu CNPJ:

```
...
new BalancoEmpresa().pagaDivida("00.000.000/0001-01", 500.0, "00.000.000/0002-02", "Empre:");
...
```

Neste caso, perceba que os 3 últimos argumentos da chamada do método `void pagaDivida(...)` representam os dados do pagamento a ser efetuado. Tanto é assim que, na implementação do método, usamos cada um destes parâmetros para preenchermos um objeto do tipo `Pagamento`.

```
public class BalancoEmpresa {
    ...
    public void pagaDivida(String cnpjCredor, double valor, String nomePagador, String cnpjPagador) {
        Divida divida = dividas.get(cnpjCredor);
        if (divida != null) {
            Pagamento pagamento = new Pagamento();
            pagamento.setCnpjPagador(cnpjPagador);
            pagamento.setPagador(nomePagador);
            pagamento.setValor(valor);
            divida.getPagamentos().registra(pagamento);
        }
    }
}
```

Vamos alterar a assinatura do método `pagaDivida(String cnpjCredor, double valor, String nomePagador, String cnpjPagador)` da classe `BalancoEmpresa` para `pagaDivida(String cnpjCredor, Pagamento pagamento)` de modo que ele possa ser usado como segue.

```
...
Pagamento pagamento1 = new Pagamento();
pagamento1.setValor(500.0);
pagamento1.setCnpjPagador("00.000.000/0002-02");
pagamento1.setPagador("Empresa Pagadora");

// esta é a maneira como desejamos poder chamar o método pagaDivida(...)
new BalancoEmpresa().pagaDivida("00.000.000/0001-01", pagamento1);
...
```

Altere a classe onde mais você achar possível usar objetos em vez de Strings. Cole aqui o código da classe

BalancoEmpresa alterado. Lembre-se que temos agora a classe Cnpj , que representa um CNPJ melhor do que uma String.

Responda

INSERIR CÓDIGO		FORMATAÇÃO
<div data-bbox="172 943 1423 1010" style="border: 1px solid #ccc; height: 30px; margin-bottom: 10px;"></div> <div data-bbox="172 1010 1423 1344" style="border: 1px solid #ccc; height: 150px;"></div>		