

PROGRAMAÇÃO EM JAVA

MÓDULO 3

TRANSCRIÇÃO DO VÍDEO

Autor: Luiz Fernando Corcini

PONTOS DE REFLEXÃO – MISSÃO 4

Vamos analisar e refletir sobre que foi feito até agora, com as classes Missao4_Conta_Corrente e Missao4_principal2:

Primeiro ponto: Criamos uma classe que tem 4 atributos privados, isto é, eles não podem ser acessados fora da classe em que eles foram criados.

- Para provar isso, faça o seguinte teste- parte 1:

Clique no final da linha 9 (depois do ponto e vírgula) e pressione ENTER;

Na linha 10 (que agora está vazia) digite “CC” (sem aspas) e logo em seguida digite ponto. Seu código deve ficar assim:



The screenshot shows an IDE with two tabs: 'Missao4_Conta_Corrente.java' and '*Missao4_principal2.java'. The code in the active tab is as follows:

```
1 package Missao4;
2
3 public class Missao4_principal2
4 {
5
6     public static void main(String[] args)
7     {
8         //criando o objeto CC baseado na classe Missao4_Conta_Corrente
9         Missao4_Conta_Corrente CC = new Missao4_Conta_Corrente();
10        CC.
11    }
12 }
13
```

A popup menu is displayed below line 10, showing a list of methods available for the 'CC' object. The methods are:

- ▲ depositar(double valor) : void - Missao4_Conta_Corrente
- equals(Object obj) : boolean - Object
- getClass() : Class<?> - Object
- hashCode() : int - Object
- notify() : void - Object
- notifyAll() : void - Object
- ▲ sacar(double valor) : void - Missao4_Conta_Corrente
- toString() : String - Object
- wait() : void - Object
- wait(long timeoutMillis) : void - Object
- wait(long timeoutMillis, int nanos) : void - Object

At the bottom of the popup, it says: 'Press 'Ctrl+Space' to show Template Proposals'.

Figura 11 - Métodos disponíveis para o objeto CC

Note, na figura acima, que, de tudo o que foi criado na Classe Missao4_Conta_Corrente (4 atributos e 2 métodos) apenas os 2 métodos: SACAR() e DEPOSITAR(), estão disponíveis. Os demais itens que aparecem nesta lista não criação nossa.

Ou seja, não temos acesso aos atributos porque eles estão definidos como privados (private).

- Para provar isso (continuar provando), faça o seguinte teste- parte 2:

Abra sua classe missao4_Conta_Corrente e troque o modificador de acesso private do atributo Nome_Cliente para “public”;

Salve o projeto;

Volte na classe Missao4_principal2 e apague o ponto colocado depois do CC (linha 10) e digite ele novamente. O menu que abriu, depois que você digitou o ponto, mostrou o atributo Nome_Cliente (destacado em azul, na figura abaixo) como primeiro item de acesso. Verdade ou Não? Código deve ter ficado assim:

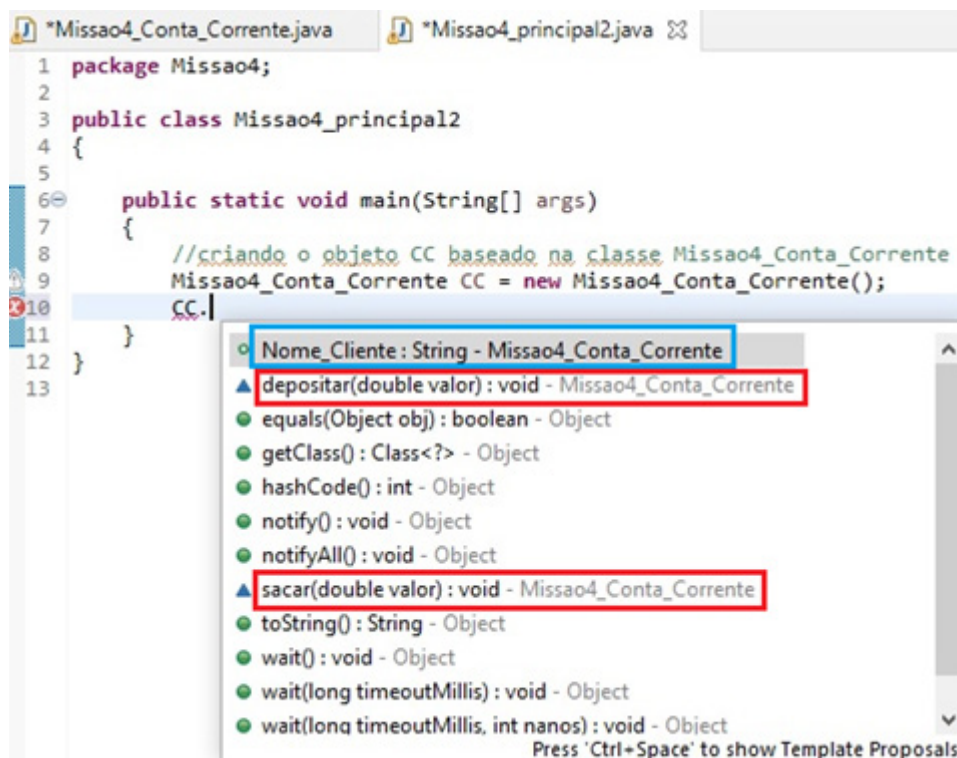


Figura 12 - Um atributo público

Entendeu agora!

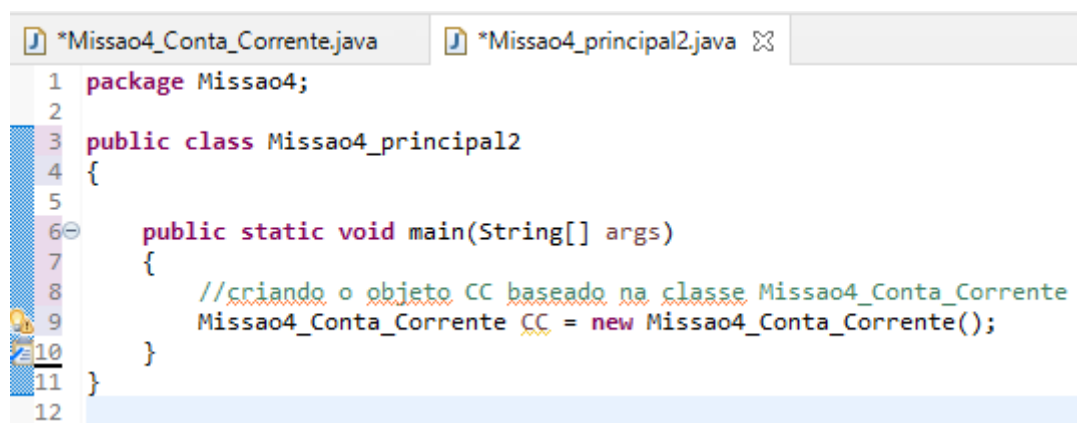
Quando o atributo é definido como público (public) ele fica acessível em outra classe (depois de instanciado o objeto), mas quando o atributo é definido como privado (private) ele fica protegido e não pode ser acessado fora da classe em que ele foi criado.

IMPORTANTE:

Lembre-se de desfazer a alteração, ou seja, retorne o atributo Nome_Cliente para private. Para ter certeza que tudo está correto, o menu de opções deve ser igual ao que aparece na figura 11;

E por que é importante que você realize os testes citados acima? É importante testar e saber o motivo das escolhas e suas funcionalidades, pois podem auxiliar na resolução de problemas. Se você não testou, volte e realize os dois testes sugeridos. Com isso você terá mais subsídios para resolução de eventuais problemas quando estiver atuando no desenvolvimento de programas em Java.

Segundo ponto: Lembra quando criamos o objeto Calculadora, na Situação de Aprendizagem 3? Pois então, quando instanciamos uma classe, criamos um objeto que podemos usar. Nós não usamos a classe (nós modelamos a classe) e, depois que ela está devidamente modelada, nós a instanciamos, criando um objeto que possuem as características (atributos) e comportamentos (métodos) definidos na classe que originou este objeto. Portanto, só conseguimos usar o objeto Calculadora depois que instanciamos a classe Missao3_Calculadora. Da mesma forma, só conseguimos usar a conta corrente (objeto CC) depois que instanciamos a classe Missao4_Conta_Corrente. Com tudo isso em mente, vamos voltar a verificar o código da figura 10, que estou colocando novamente aqui abaixo, para facilitar:



```
1 package Missao4;
2
3 public class Missao4_principal2
4 {
5
6     public static void main(String[] args)
7     {
8         //criando o objeto CC baseado na classe Missao4_Conta_Corrente
9         Missao4_Conta_Corrente CC = new Missao4_Conta_Corrente();
10    }
11 }
12
```

Figura 13 - Cópia da figura 10

Perceba que, se você estivesse abrindo uma conta (de verdade) num banco (de verdade) quando o compilador executasse a linha 9 desta figura:

```
Missao4_Conta_Corrente CC = new Missao4_Conta_Corrente();
```

Estaríamos criando uma conta corrente de verdade! Esta conta corrente seria um objeto (baseado na classe Missao4_Conta_Corrente e seria chamado de “CC”.

Então, seria muito legal que, no instante que esta conta fosse criada, uma mensagem informando este fato fosse exibida no console, para termos certeza que a conta foi criada com sucesso, não é mesmo?