

Criando as classes

Esta é a classe de domínio, aquela que possui atributos e métodos, neste caso, apenas o método chamado `alterarDoisNumeros` recebendo dois parâmetros inteiros `numero1` e `numero2`.

```
public class Calculadora {  
    public void alterarDoisNumeros( int numero1, int numero2 ) {  
        numero1 = 99;  
        numero2 = 33;  
    }  
}
```

Uma cópia dos valores são recebidos, dentro do escopo são alterados sem afetar os valores originais lá dentro do main.

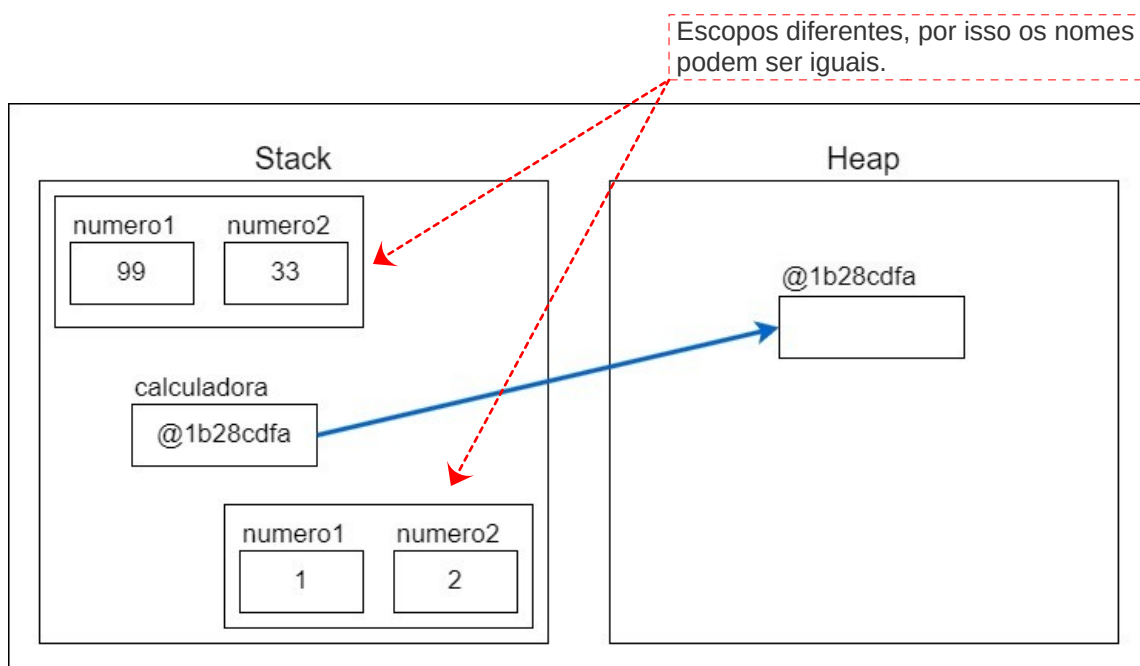
Já esta é a classe que possui o método main, aquela utilizada para criar as instâncias das classes de domínio, como a classe de domínio `Calculadora` que foi criada.

```
public class CalculadoraTeste {  
    public static void main(String[] args) {  
        Calculadora calculadora = new Calculadora();  
        int numero1 = 1;  
        int numero2 = 2;  
        calculadora.alterarDoisNumeros( numero1, numero2 );  
    }  
}
```

Passa uma cópia do valor para o método, dentro dele é alterado sem afetar o valor original. Não importa o valor passado, dentro do método será alterado, pois é enviado uma cópia do valor para o método.

Enquanto isso, na memória do computador...

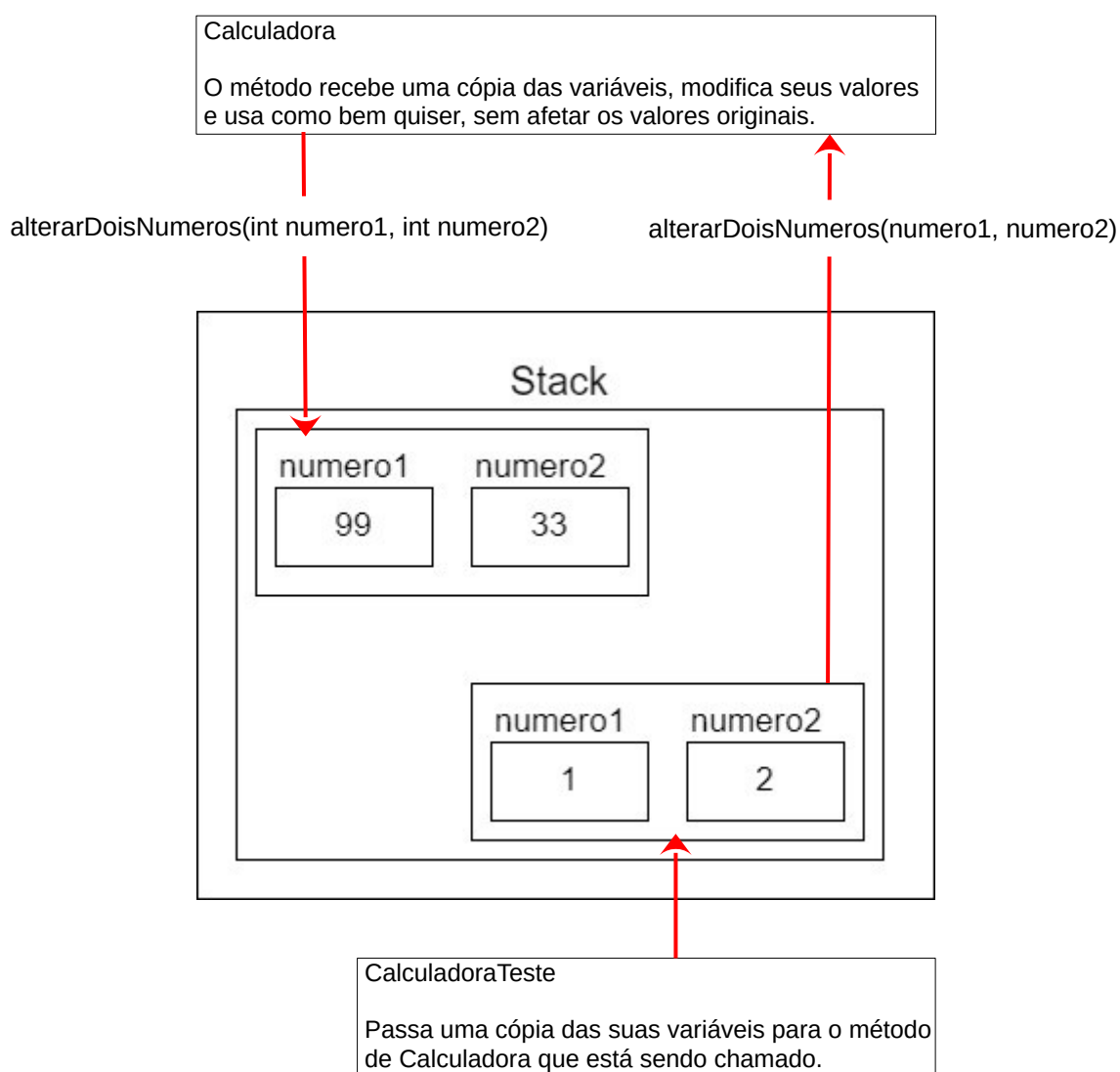
Esta é a representação do código acima funcionando na memória do computador, como se pode ver, tudo está acontecendo em apenas um espaço na memória, o espaço conhecido como Stack, mas sabe por que isso acontece? Porque é nessa parte que as variáveis primitivas e de escopo são criadas. É por isso que os valores foram diferentes, conforme mostrado na aula, [confira](#).



Sabia que variáveis não podem ser criadas com mesmo nome, né? Mas quando elas estão em **escopos diferentes** isso é perfeitamente possível, na memória elas são representadas como uma caixinha e dentro dela estarão as **variáveis locais** enquanto o escopo onde foram criadas existir.

Viu que eu também criei uma variável local chamada `calculadora`? Essa é do tipo Calculadora que guarda apenas o endereço de um objeto fazendo referência a ele, nada mais.

Mas como funciona essa questão de parâmetro por valor?



Se os valores de cada variável forem impressos serão diferentes, pois o **CalculadoraTeste** passa uma cópia dos valores de suas variáveis (1 e 2), porém dentro do método lá em **Calculadora** cada valor é alterado (99 e 33) e isso não afeta o valor original passado pela classe **CalculadoraTeste**.

É assim que funciona os parâmetros por valor, significa que uma cópia dos valores são passados como argumento, esses valores podem ser alterados no método sem afetar os valores originais na classe que passou os valores no argumento.