# Modificadores de Acesso

Os modificadores de acesso são padrões de visibilidade de acessos às classes, atributos e métodos. Esses modificadores são palavras-chaves reservadas pelo Java, ou seja, palavras reservadas não podem ser usadas como nome de métodos, classes ou atributos.

Como boas práticas (*best practices*) do Java, na maioria das declarações de variáveis de instância são definidos os seus atributos com a palavra-chave **private**, para garantir a segurança de alterações acidentais, sendo somente acessíveis através dos métodos. Essa ação tem como efeito ajudar no encapsulamento dos dados, preservando ainda mais a segurança e a aplicação de programação orientada a objetos do Java.

Por exemplo, quando um programa cria (instancia) um objeto da classe Banco, a variável senha é encapsulada (ocultada) no objeto onde pode ser acessada apenas por métodos da classe do objeto, os métodos **getters** e **setters**, que manipulam a variável de instância. Vejamos abaixo a explicação sobre cada um.

## Public:

Uma declaração com o modificador **public** pode ser acessada de qualquer lugar e por qualquer entidade que possa visualizar a classe a que ela pertence.

## Private:

Os membros da classe definidos como **private** não podem ser acessados ou usados por nenhuma outra classe. Esse modificador não se aplica às classes, somente para seus

métodos e atributos. Esses atributos e métodos também não podem ser visualizados pelas classes herdadas.

## Protected:

O modificador **protected** torna o membro acessível às classes do mesmo pacote ou através de herança, seus membros herdados não são acessíveis a outras classes fora do pacote em que foram declarados.

## Default (padrão):

A classe e/ou seus membros são acessíveis somente por classes do mesmo pacote, na sua declaração não é definido nenhum tipo de modificador, sendo este identificado pelo compilador.

**Listagem 1:** Exemplo de declaração do modificador default

```
1 public class Modificador_Default {
2
3         public static void main(String[] args) {
4
5                 String nome = "Flávia Bernandes";
6                 System.out.printf("Nome.: %s", nome);
7         }
8 }
```

## Final:

Quando **final** é aplicado na classe este não permite estende-la. Nos métodos impede que o mesmo seja sobrescrito (*overriding*). Na subclasse e nos valores de variáveis não podem ser alterados depois que já tenha sido atribuído um valor.

## Abstract:

Esse modificador não é aplicado nas variáveis, apenas nas classes. Uma classe abstrata não pode ser instanciada, ou seja, não pode ser chamada pelos seus construtores. Se houver alguma declaração de um método como **abstract** (abstrato), a classe também deve ser marcada como abstract.

## Static:

É usado para a criação de uma variável que poderá ser acessada por todas as instâncias de objetos desta classe como uma variável comum, ou seja, a variável será a mesma em todas as instâncias e quando seu conteúdo é modificado numa das instâncias, a modificação ocorre em todas as demais. E nas declarações de métodos ajudam no acesso direto à classe, portanto não é necessário instanciar um objeto para acessar o método.

No exemplo da Listagem 2, é gerada a saída da contagem de 4 valores (1,2,3,4), se comentarmos a linha que tem a variável **static** e descomentar da qual não tem, veremos que a sua saída será de 4 saídas com o mesmo valor (1,1,1,1).

Nas variáveis estáticas isso acontece porque todas as instâncias da mesma classe compartilham a mesma cópia das variáveis estáticas, sendo inicializadas quando a classe é carregada (instanciada).

**Listagem 2:** Exemplo do modificador static

```
1  public class Conta_Instancias {
2          private int tamanho;
3          private static int conta = 0; //IMPRIME A CONTAGEM DE 4 VALORES
4          //private int conta = 0; //IMPRIME A CONTAGEM DE 1 VALOR
5
6          public Conta_Instancias(){
7                  conta++;
8                  System.out.println("Valor = "+conta);
9          }
10
11         public static void main(String[] args) {
12                 Conta_Instancias c = new Conta_Instancias();
13                 Conta_Instancias dois = new Conta_Instancias();
14                 Conta_Instancias tres = new Conta_Instancias();
15                 Conta_Instancias quatro = new Conta_Instancias();
16         }
17
18 }
```

## Tabela dos modificadores de acesso

| | private | default | protected | public |
|---|---|---|---|---|
| **Mesma classe** | Sim | Sim | Sim | Sim |
| **Mesmo pacote** | Não | Sim | Sim | Sim |
| **Pacotes diferentes (subclasses)** | Não | Não | Sim | Sim |
| **Pacotes diferentes (sem subclasses)** | Não | Não | Não | Sim |

*no Java ainda existem alguns outros tipos que não são muito usados