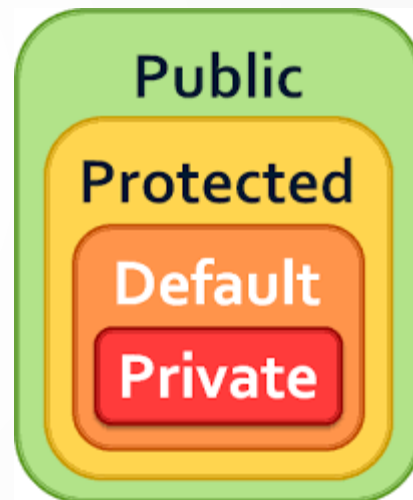
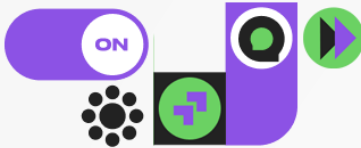


Modificadores de acesso

São expressões que o objetivo de tornar componentes da aplicação menos ou mais acessíveis por outras partes do seu programa.

São 4 os modificadores de acesso básicos da linguagem Java: *private*, *padrão*, *protected* e *public*.





Public

O modificador de acesso public é o menos restritivo de todos. Ele permite que qualquer outra parte da sua aplicação tenha acesso ao componente marcado como public.



Protect

Os membros das classes marcados com o modificador de acesso protected serão acessíveis por classes e interfaces dentro do mesmo pacote e por classes derivadas mesmo que estejam em pacotes diferentes.



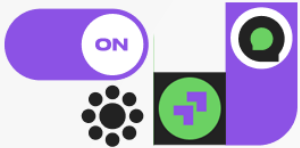
Default

O modificador de acesso padrão, é o modificador atribuído aos membros da classe que não foram marcados explicitamente com um outro modificador de acesso. Membros com acessibilidade de pacote só podem ser acessados por outras classes ou interfaces definidas dentro do mesmo pacote.



Private

Private é o mais restritivo entre os modificador de acesso. Todo membro de uma classe definido com o modificador private só é acessível para a própria classe. Não importa a localização dentro de pacotes ou se a classe foi herdada ou não, um membro private só é acessível dentro da mesma classe em que ele foi declarado.



VISIBILIDADE	public	protect	default	private
A partir da mesma classe	x	x	x	x
Qualquer classe do mesmo pacote	x	x	x	
Qualquer classe filha, no mesmo pacote	x	x	x	
Qualquer classe filha em pacote diferente	x	x		
Qualquer classe de pacote diferente	x			

Classes

Classes são elementos de código Java que usamos para representar objetos do mundo real. Nelas, geralmente são declaradas propriedades (atributos) e métodos, que representam as características e o comportamento do objeto. Podemos pensar que as classes é uma forma de agrupar atributos e métodos de um mesmo contexto.

Uma classe deve ser coesa, ou seja, possuir uma única responsabilidade. Ela não toma conta de mais de um conceito no sistema. Se a classe é responsável por representar uma Fatura, ela representa apenas isso. As responsabilidades de detalhar Compras, por exemplo, estarão em outra classe.

Sintaxe:

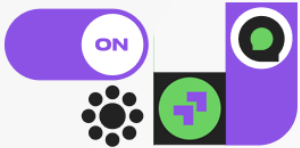
```
<modificador de acesso> class NomeDaClasse {
```

```
    // Local onde construtores, atributos, e métodos são determinados
```

```
}
```

Classes

Classes coesas são vitais em um sistema orientado a objetos. Elas são mais simples de serem mantidas, possuem menos código e seu reúso é maior.

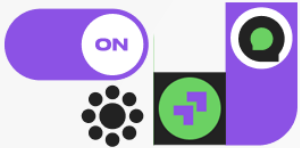


```
public class Cliente {  
  
    private String nome;  
    private int idade;  
  
    public Cliente() {  
        // Código do construtor  
    }  
  
    public void buscarNomeCliente() {  
        // Código do método  
    }  
  
}
```

Regras para nomeação de classes

Ao nomear uma classe algumas convenções devem ser seguidas:

1. Classes utilizamos substantivos e para funções verbos
2. Manter o nome simples e descritivo;
3. Usar palavras inteiras, isto é, sem siglas e abreviações;
4. A primeira letra de cada palavra deve estar em caixa alta.
(CamelCase)



```
public class Cliente {
```

```
    private String nome;  
    private int idade;
```

```
    public Cliente() {  
        // Código do construtor  
    }
```

```
    public void buscarNomeCliente() {  
        // Código do método  
    }
```

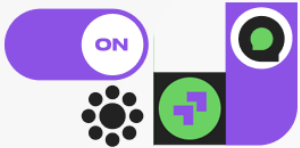
```
}
```



Atributos

Os atributos são as propriedades de um objeto, também são conhecidos como variáveis de classe.

Essas propriedades fazem parte da estrutura da classe que define o estado de um objeto. Elas são compartilhadas por todos os objetos gerados a partir dessa classe, porém com valores distintos e que podem sofrer alterações.



```
public class Cliente {
```

```
    private String nome;  
    private int idade;
```

```
    public Cliente() {  
        // Código do construtor  
    }
```

```
    public void buscarNomeCliente() {  
        // Código do método  
    }
```

```
}
```

Construtores

São os responsáveis por criar o objeto em memória, ou seja, instanciar a classe que foi definida. Por padrão, o Java cria construtor sem parâmetros para todas as classes. Por outro lado, se você quiser, poderá criar mais de um construtor para uma mesma classe, passando quantos atributos necessários.

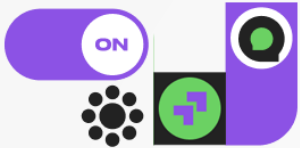
Métodos

Um método é um bloco de código que somente é executado quando é chamado, nele você pode passar parâmetros, e utilizá-los para desenvolver comportamentos.

Os métodos são usados para executar determinadas ações e também são conhecidos como funções.

Por que criar métodos ao invés de reescrever os comportamentos desejados?

Para poder reutilizar pedaços código, ganhando em tempo, desempenho e manutenibilidade, pois uma vez definido o trecho de código, use-o quantas vezes necessárias.



```
public class Cliente {  
  
    private String nome;  
    private int idade;  
  
    public Cliente() {  
        // Código do construtor  
    }  
  
    public void buscarNomeCliente() {  
    }  
    static void meuMetodoStatic() {  
        System.out.println("Métodos estáticos  
podem ser chamados sem criar objetos");  
    }  
}
```

Métodos

Métodos *Static*, significa que ele pode ser acessado sem criar um objeto da classe, diferente public que só pode ser acessado por objetos.

Links Úteis

ACOPLAMENTO, COESÃO E ENCAPSULAMENTO EM ORGANIZAÇÕES

<https://targetteal.com/pt/blog/acoplamento-coesao-encapsulamento/>

Obrigada



