

Modificadores de Acesso e Construtores



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Modificadores de Acesso
- Public
- Protected
- Private
- Default
- Construtores

Modificadores de Acesso

No Java nós temos forma de limitar quem pode acessar uma classe, essa modificação se dá por meio dos modificadores de acesso. Eles podem ser **public**, **protected**, **private** e **default**. Cada um desses modificadores limita o acesso a uma classe de forma diferente.

Esse acesso não se limita apenas à classe em si, mas também aos Métodos e Atributos que existem dentro da classe em questão. Podemos ter uma classe como public e seus atributos como private ou protected. Vamos ver mais sobre isso a seguir.

O modificador de acesso *Public* permite que a classe possa ser acessada por qualquer outra classe no nosso programa. Sendo assim, podemos criar objetos para ela à vontade e também podemos acessar todos os métodos dessa classe pelos objetos criados com ela.

Os métodos que estão dentro dessa classe podem ter outro modificador de acesso

Protected

O modificador Protected limita o acesso de uma classe apenas aos filhos dela e a classes que estejam no mesmo pacote(a mesma pasta), sendo assim não é qualquer classe que pode ter acesso a um atributo ou método Protected.

Classes no mesmo pacote podem acessar os métodos ou atributos dessa classe, mas outras classes não vão nem ter o acesso a existência do que tem dentro dessa classe.

O Modificador Private pode ser pensado como apenas visível dentro da própria classe. Uma classe Private que tenha atributos não poderão ser acessados por nenhuma outra, da mesma forma métodos private não podem ser acessados por outras classes.

O Private isola um atributo ou método do resto do programa.

Podemos pensar no Private como: Apenas visível dentro da própria classe

Default

O modificador *Default* é utilizado por padrão quando criamos uma classe sem modificadores. Ele permite que uma classe seja acessada dentro do pacote no qual ela se encontra. O Default pode ser aplicado tanto para classes quanto para métodos e para atributos também.

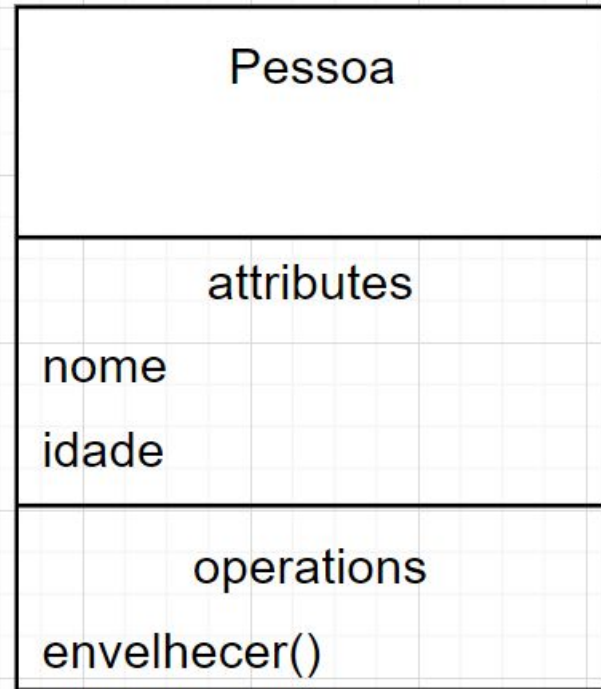
Modificadores de Acesso

	default	private	protected	public
Same Class	Yes	Yes	Yes	Yes
Same package subclass	Yes	No	Yes	Yes
Same package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

[Access Modifiers in Java - GeeksforGeeks](#)

Modificadores de Acesso

As Classe podem ser apenas Public e Default.



Mão na massa

Vamos fazer algumas classes e testar esses modificadores para vermos os seus comportamentos.

INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:40



Construtores são uma forma de *construir* um objeto novo. Até agora a única forma que tínhamos de adicionar informações a um objeto era através de Setters e métodos. Agora poderemos criar um objeto novo já com as informações que queremos.

Vamos ver como podemos fazer isso e quais são as vantagens dos construtores.

Construtores

Vamos criar um construto para uma classe Pessoa que tenha 2 atributos, nome e idade:

```
public class Pessoa {  
    private String nome;  
    private Integer idade;
```

```
    public Pessoa(String nome, Integer idade) {  
        this.nome = nome;  
        this.idade = idade;  
    }  
}
```

Construtores

Como podemos ver no exemplo anterior, o Construtor parece muito com um método, porém ele tem que ter o mesmo nome da classe e não tem um tipo de retorno. A ideia de um construtor é receber os valores que equivalem aos atributos da classe e vai atribuí-los ao objeto.

Podemos ter também um construtor que não receba nenhum valor

```
public Pessoa() {  
  
}
```

Esse construtor se chama construtor padrão.

Podemos ter vários construtores por classe, sendo assim podemos ter um construtor padrão, e um construtor parametrizado na mesma classe. Podemos também ter outro construtor parametrizado, contanto que ele receba parâmetros diferentes.

Sendo assim podemos pensar em diversos cenários de criação de um mesmo objeto, apenas mudando os valores que ele pode receber.

Construtores

```
public Pessoa(){  
}  
public Pessoa(String nome, Integer idade){  
    this.nome = nome;  
    this.idade = idade;  
}  
public Pessoa(String nome, Integer idade, String cpf){  
    this.nome = nome;  
    this.idade = idade;  
    this.cpf = cpf;  
}  
public Pessoa(Integer idade, String cpf){  
    this.idade = idade;  
    this.cpf = cpf;  
}  
public Pessoa(String cpf){  
    this.cpf = cpf;  
}
```


AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>