

Modificadores de acesso em Java



Regiane Moreno





CONCEITO

Conceito

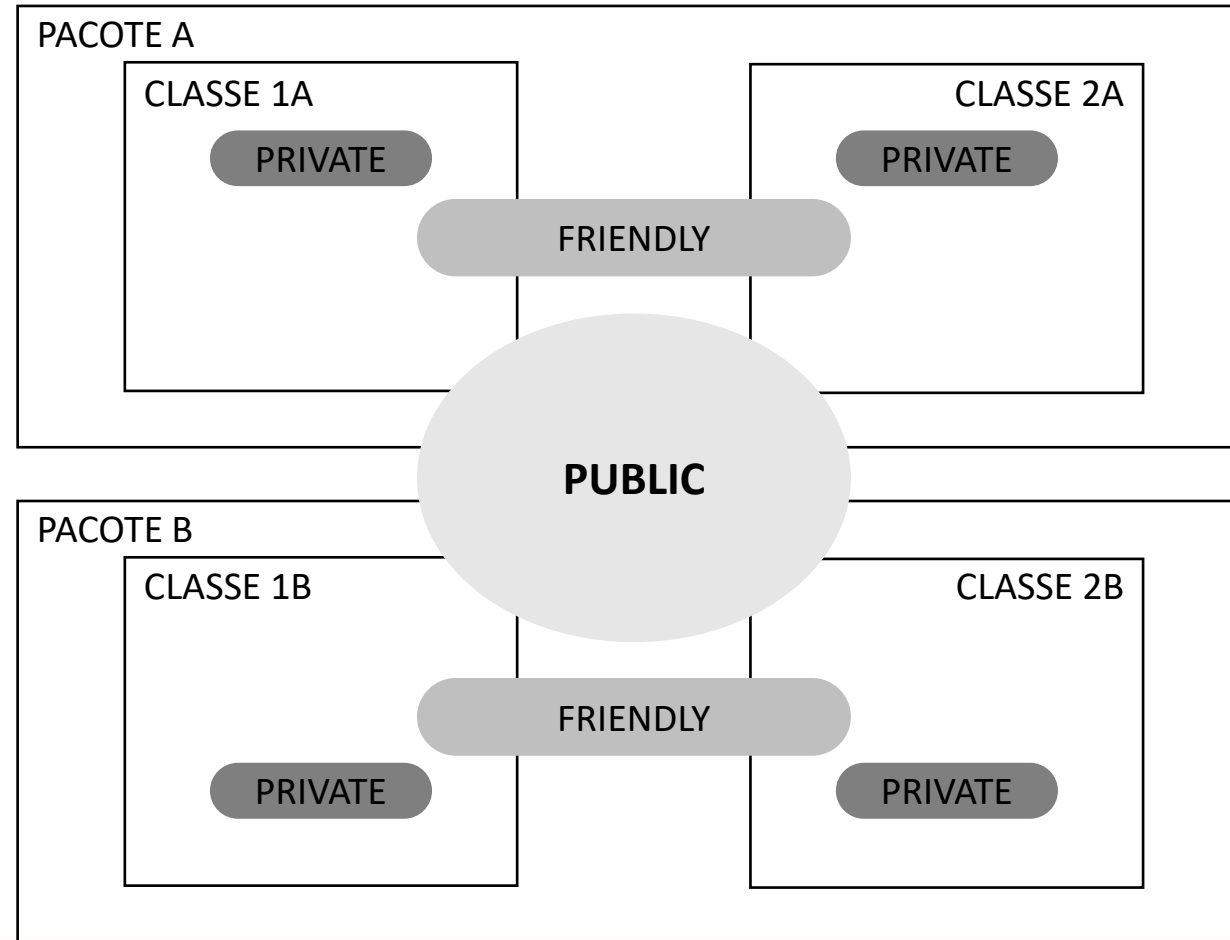
MODIFICADOR	DESCRIÇÃO
PUBLIC	Uma declaração com o modificador public pode ser acessada de qualquer lugar e por qualquer entidade que possa visualizar a classe a que ela pertence.
PRIVATE	Os membros da classe definidos como private não podem ser acessados ou usados por nenhuma outra classe. Esse modificador 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.
PADRÃO(DEFAULT)	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 computador. Chamados de " <i>package</i> " ou " <i>friendly</i> ".



VISIBILIDADE

Visibilidade

Graficamente:



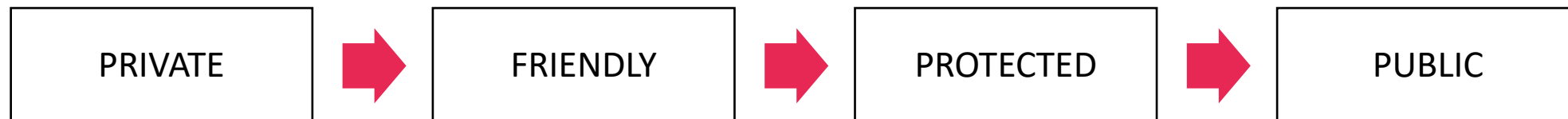


ACESSIBILIDADE

Acessibilidade

MODIFICADOR	CLASSE	MESMO PACOTE	PACOTE DIFERENTE (SUBCLASSE)	PACOTE DIFERENTE (GLOBAL)
Public	✓	✓	✓	✓
Protected	✓	✓	✓	✗
Default	✓	✓	✗	✗
Private	✓	✗	✗	✗

Assim, temos do modificador de acesso MAIS restritivo para o MENOS restritivo:





EXEMPLO SEM ENCAPSULAMENTO E EXEMPLO COM ENCAPSULAMENTO

Exemplo sem encapsulamento

```
class ContaCorrente {  
    String nome;  
    double saldo;  
  
    ContaCorrente (double saldo){  
        this.saldo = saldo;  
    }  
  
    void deposito (double valor) {  
        saldo = saldo + valor;  
    }  
  
    void saque (double valor) {  
        saldo = saldo - valor;  
    }  
}
```

```
.  
.public static void main(String [] args) {  
    ContaCorrente cta;  
    cta = new Contacorrente(1000);  
  
    cta.deposito(200);  
    cta.saque (100);  
    cta.saldo = 5000;  
}
```

Por meio da
instância, é
possível alterar
qualquer
atributo.

Exemplo com encapsulamento

```
class ContaCorrente {  
    private String nome;  
    private double saldo;  
  
    ContaCorrente (double saldo){  
        this.saldo = saldo;  
    }  
  
    public void deposito (double valor) {  
        saldo = saldo + valor;  
    }  
  
    public void saque (double valor) {  
        saldo = saldo - valor;  
    }  
}
```

```
.  
.  
public static void main(String []  
args) {  
  
    ContaCorrente cta;  
    cta = new ContaCorrente(1000);  
    cta.deposito(200);  
    cta.saque (100);  
    cta.saldo = 5000;  
}
```



Erro!



GETTERS E SETTERS

Getters e Setters

Há necessidade de se ter métodos públicos para que se possa acessar os atributos que, em geral, são privados. O padrão adotado, pelos programadores em Java, para estes métodos é **setNomeAtributo()** e **getNomeAtributo()** para modificar e receber os valores dos atributos, respectivamente.

Estes métodos são comumente chamados de **Getters** e **Setters**. Um atributo privado pode ter seu conteúdo alterado (set) e recuperado (get) por meio destes métodos.

Métodos de acesso ➔ `getXXX()` – permitem o acesso a algum atributo de uma classe.

Métodos modificadores ➔ `setXXX()` – alteram algum atributo de uma classe.

Exemplo com Getters e Setters

```
public class ContaCorrente {  
    private String nome;  
    private double saldo;  
  
    ContaCorrente (double saldo){  
        this.saldo = saldo;  
    }  
  
    public void deposito (double valor) {  
        setSaldo(getSaldo() + valor);  
    }  
  
    public void saque (double valor) {  
        setSaldo(getSaldo() - valor);  
    }  
}
```

```
    public String getNome() {  
        return nome;  
    }  
  
    public void setNome(String nome) {  
        this.nome = nome;  
    }  
  
    public double getSaldo() {  
        return saldo;  
    }  
  
    public void setSaldo(double saldo) {  
        this.saldo = saldo;  
    }  
}
```

```
public static void main(String[] args) {  
    ContaCorrente cta;  
    cta = new ContaCorrente(1000.00);  
    cta.deposito(200);  
    cta.saque (100);  
    cta.setSaldo(5000.00);  
    System.out.println("Saldo:" + cta.getSaldo());  
  
}  
  
}
```

