

## TEMA 10: MODIFICADORES DE ACESSO EM JAVA

Em Java, os modificadores de acesso **determinam a visibilidade de classes, variáveis, métodos e construtores**. Há quatro tipos de modificadores de acesso:

1. **private**: O membro é acessível apenas dentro da sua própria classe.
2. **default** (sem modificador): O membro é acessível dentro do mesmo pacote. Este é usado se você não especificar um modificador.
3. **protected**: O membro é acessível dentro do seu próprio pacote e por subclasses.
4. **public**: O membro é acessível de qualquer outra classe.

### 1. Classe Livro no pacote biblioteca:

```
// Livro.java
package biblioteca;

public class Livro {
    private String autorPrivado = "Autor Privado";
    String autorDefault = "Autor Default"; // modificador default
    protected String autorProtegido = "Autor Protegido";
    public String autorPublico = "Autor Público";

    public String getAutorPrivado() {
        return autorPrivado;
    }
}
```

### 2. Classe BibliotecaTeste no pacote biblioteca:

```
// BibliotecaTeste.java
package biblioteca;

public class BibliotecaTeste {
    public static void main(String[] args) {
        Livro livro = new Livro();

        System.out.println("Dentro do mesmo pacote (biblioteca):");
        System.out.println("Private: " + livro.getAutorPrivado()); // Acesso via método público
        System.out.println("Default: " + livro.autorDefault);
        System.out.println("Protected: " + livro.autorProtegido);
        System.out.println("Public: " + livro.autorPublico);
    }
}
```

### 3. Classe LivroEspecial fora do pacote biblioteca, mas estendendo Livro:

```
// LivroEspecial.java
package lojadelivros;

import biblioteca.Livro;
```

```

public class LivroEspecial extends Livro {
    public void mostrarAutores() {
        // Não pode acessar autorPrivado porque é private
        // Não pode acessar autorDefault porque é default e estamos em um pacote diferente
        System.out.println("Protected (via subclasse em pacote diferente): " + autorProtegido);
        System.out.println("Public (via subclasse): " + autorPublico);
    }
}

```

#### 4. Classe Loja fora do pacote biblioteca:

```

// Loja.java
import biblioteca.Livro;
import lojadelivros.LivroEspecial;

public class Loja {
    public static void main(String[] args) {
        Livro livro = new Livro();

        System.out.println("Fora do pacote:");
        // Não pode acessar autorPrivado porque é private
        // Não pode acessar autorDefault porque é default e estamos fora do pacote
        // Não pode acessar autorProtegido porque não é subclasse e estamos fora do pacote
        System.out.println("Public: " + livro.autorPublico);

        LivroEspecial livroEspecial = new LivroEspecial();
        livroEspecial.mostrarAutores();
    }
}

```

#### Saídas esperadas:

##### Ao executar BibliotecaTeste:

Dentro do mesmo pacote (biblioteca):

Private: Autor Privado

Default: Autor Default

Protected: Autor Protegido

Public: Autor Público

##### Ao executar Loja:

Fora do pacote:

Public: Autor Público

Protected (via subclasse em pacote diferente): Autor Protegido

Public (via subclasse): Autor Público

**Vamos associar as situações baseando-se no exemplo:**

**1. private:** O membro é acessível apenas dentro da sua própria classe.

Código Responsável:

```
private String autorPrivado = "Autor Privado";
```

Na classe Livro, temos o atributo autorPrivado definido com o modificador private. A única maneira de acessar este atributo de fora da classe Livro é através de um **método público**, como o getAutorPrivado().

**2. default** (sem modificador): O membro é acessível dentro do mesmo pacote. Este é usado se você não especificar um modificador.

Código Responsável:

```
String autorDefault = "Autor Default";
```

Na classe Livro, temos o atributo autorDefault que está usando o modificador default (**sem modificador explícito**). Ele é acessível diretamente pela classe BibliotecaTeste porque ambas estão no mesmo pacote biblioteca.

**3. protected:** O membro é acessível dentro do seu próprio pacote e por subclasses.

Código Responsável:

```
protected String autorProtegido = "Autor Protegido";
```

Na classe Livro, temos o atributo autorProtegido definido com o **modificador protected**. Ele é acessível tanto pela classe BibliotecaTeste (pois está no mesmo pacote) quanto pela classe LivroEspecial (porque LivroEspecial é uma subclasse de Livro).

**4. public:** O membro é acessível de qualquer outra classe.

Código Responsável:

```
public String autorPublico = "Autor Público";
```

Na classe Livro, temos o atributo autorPublico definido com o **modificador public**. Este atributo é acessível por qualquer outra classe, independentemente do pacote em que ela esteja. Portanto, classes como BibliotecaTeste, LivroEspecial e Loja podem acessá-lo diretamente.