

Encapsulamento

Definição

- **Encapsulamento:** técnica utilizada para restringir o acesso a variáveis (atributos), métodos ou até a própria classe.
- Os detalhes da implementação ficam ocultos ao usuário da classe, ou seja, o usuário passa a utilizar os serviços (métodos) de uma classe sem saber como isso é feito internamente.

Controles de Acesso

- Existem os seguintes controles de acesso:
 - public: sem restrições, equivalente a não encapsular.
 - private: nível de mais restritivo, apenas métodos da própria classe tem acesso a variáveis e métodos privados.
 - protected: nível intermediário de encapsulamento em que variáveis e métodos podem ser acessados pela própria classe ou por suas subclasses.

Controles de Acesso para Classes

- Uma classe pode ter dois controles de acesso:
 - public: a classe é acessível por qualquer outra classe.
 - private: a classe é acessível apenas por classes do mesmo pacote. Não é colocada a palavra private na declaração da classe, isso é feito de forma implícita.

Controles de Acesso para Classes

Exemplo

- Classe privada teste no pacote introducao_oo:

```
package introducao_oo;  
  
class teste {  
    int a, b;  
}
```

- Classe teste_qualificador, do pacote introducao_oo_teste_qualificador tentado incluir a classe teste, mas não é possível, pois ela é privada e está num pacote diferente.

```
package introducao_oo_teste_qualificador;  
  
import introducao_oo.Produto1;  
//import introducao_oo.teste;  
  
public class teste_qualificador {  
  
    public static void main(String[] args) {  
    }  
}
```

Controles de Acesso para Classes

Exemplo

- Classe privada teste2 no pacote interfaces:
- Classe TesteInterfaces, do mesmo pacote interfaces, acessando a classe privada teste2.

```
package interfaces;
```

```
class teste2 {
```

```
}
```

```
public class Pessoa {
```

```
    private String nome = null;
```

```
}
```

```
package interfaces;
```

```
public class TesteInterfaces {
```

```
    public static void main(String[] args) {
```

```
        teste2 t = new teste2();
```

```
    }
```

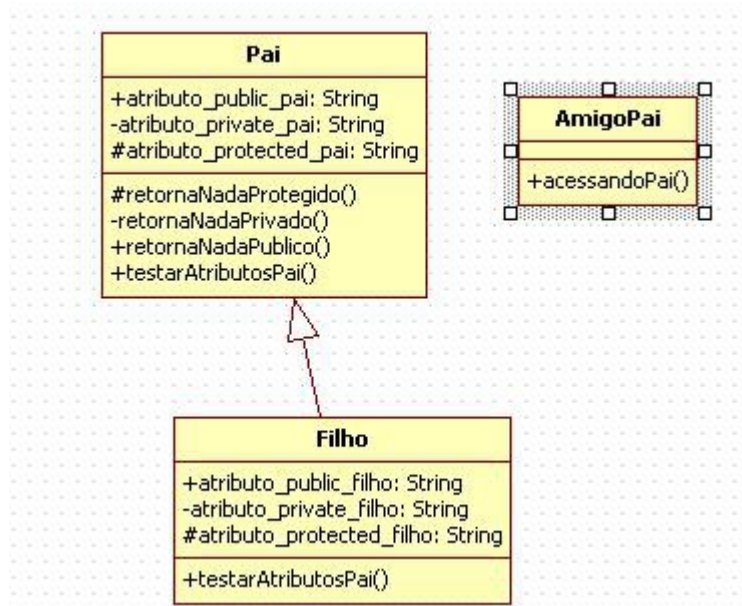
```
}
```

Controles de acesso para atributos e métodos

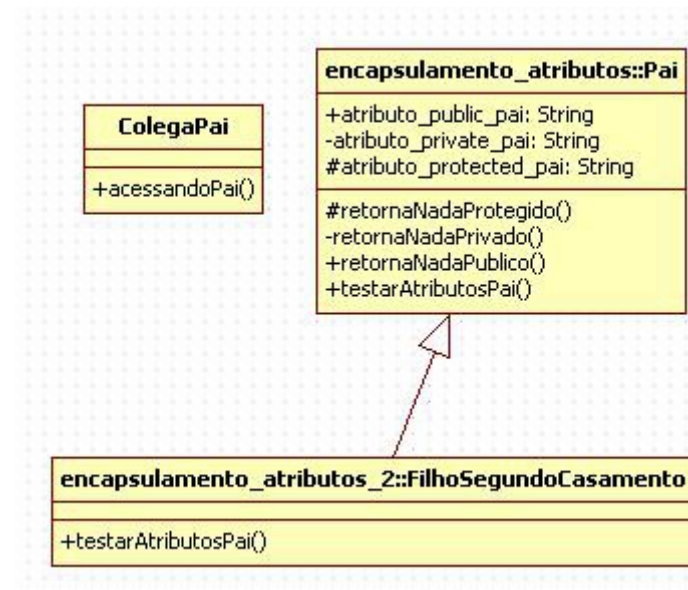
- Um atributo ou método pode ter os 3 níveis de acesso.
 - public: atributos e métodos sempre acessíveis.
 - private: atributos e métodos acessíveis apenas para métodos da classe.
 - protected: atributos e métodos acessíveis apenas para métodos da classe, das subclasses e de classes do mesmo pacote.

Pacotes

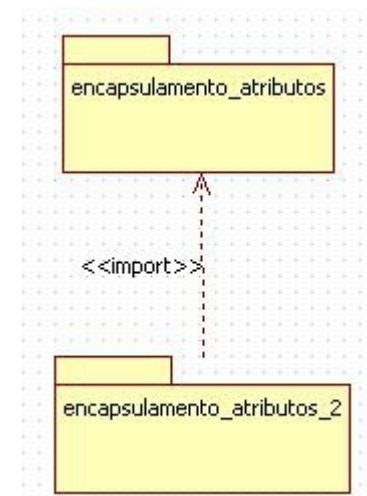
- encapsulamento_atributos



- encapsulamento_atributos_2



- pacotes



Controles de acesso para atributos Ex:

```
package encapsulamento_atributos;

public class Pai {

    public String atributo_public_pai;

    private String atributo_private_pai;

    protected String atributo_protected_pai;

    public void testarAtributosPai() {

        this.atributo_private_pai = "Acessando
        privado";

        this.atributo_protected_pai = "Acessando
        protegido";

        this.atributo_public_pai = "Acessando
        público";

    }

}
```

```
package encapsulamento_atributos;

public class Filho extends Pai {

    public void testarAtributosPai() {

        // this.atributo_private_pai = "Acessando privado";

        this.atributo_protected_pai = "Acessando
        protegido";

        this.atributo_public_pai = "Acessando público";

    }

}
```

Obs. Não é possível acessar a um atributo privado.

Controles de acesso para atributos Ex:

```
package encapsulamento_atributos;

public class AmigoPai {

    public void acessandoPai(){

        Pai pai = new Pai();

        pai.atributo_protected_pai = "Amigo";

        pai.atributo_public_pai = "Amigo";

    }

}
```

Obs. O AmigoPai está no mesmo pacote que o Pai por isso consegue acessar o atributo protegido do Pai

```
package encapsulamento_atributos_2;

import encapsulamento_atributos.Pai

public class FilhoSegundoCasamento extends Pai {

    public void testarAtributosPai() {

        // this.atributo_private_pai = "Acessando
        //   privado";

        this.atributo_protected_pai = "Acessando
        protegido";

        this.atributo_public_pai = "Acessando
        público";

    }

}
```

Obs. O FilhoSegundoCasamento consegue acessar o atributo protegido porque apesar de estar em outro pacote descende do Pai.

Controles de acesso para atributos Ex:

```
package encapsulamento_atributos_2;
```

```
import encapsulamento_atributos.Pai;
```

- Note que não é possível acessar o atributo protegido

```
public class ColegaPai {
```

```
    public void acessandoPai(){
```

```
        Pai pai = new Pai();
```

```
        pai.atributo_public_pai = "Colega";
```

```
//        pai.atributo_protected_pai = "Amigo";
```

```
    }
```

```
}
```

Controles de acesso para métodos

- Os controles de acesso para métodos se comportam da mesma forma que o controle de acesso para atributos.
- Crie um método público, um método privado e um método protegido na classe Pai. Tente acessar esses métodos nas classes Filho, AmigoPai, FilhoSegundoCasamento.

Dúvidas?

