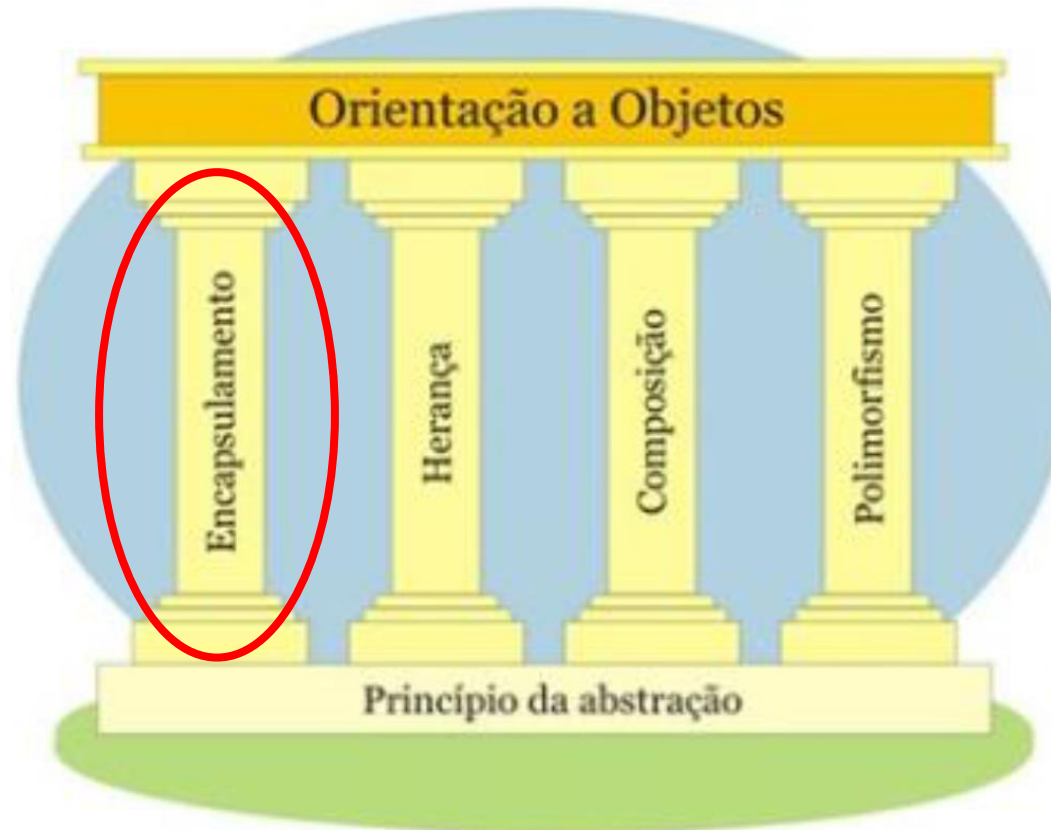


Pilares POO



Encapsulamento



Disponibiliza métodos que operam sobre os dados e que protegem o acesso direto indevido aos atributos de uma instância fora da classe onde estes foram declarados.

Encapsulamento

- Um objeto, em um programa, “encapsula” todo o seu **estado** e o **comportamento**
- Os dados e as operações são agrupados e a sua implementação é escondida, protegida dos usuários

Proteção de Dados

Visa garantir o acesso sobre operações e atributos disponibilizados pela interface da classe. Os modificadores de acesso são:

- **Public**: uma classe declarada como pública dá a outras classes, de todos os pacotes, acesso aos seus membros que forem públicos
- **Protected**: quando um membro de uma classe é declarado assim, ele torna acessível por classes através de herança.
- **Private**: o membro de classe não pode ser acessado por nenhuma outra classe.

	private	default	protected	public
mesma classe	sim	sim	sim	sim
mesmo pacote	não	sim	sim	sim
pacotes diferentes (subclasses)	não	não	sim	sim
pacotes diferentes (sem subclasses)	não	não	não	sim

Proteção de Dados

Todos os atributos e operações de uma classe podem ser acessados pelas operações da mesma classe.

O acesso aos **atributos** é, em geral, **privado** ou **protegido**.

```
obj.saldo = 10000000; //não se acessa atributo privado assim!
```

O acesso às **operações** que fazem parte da interface da classe é **público**.

```
obj.alterarSaldo(1000); //é possível se alterar um método público fora da classe
```

Métodos Get e Set

- Veja na figura que representa a classe Aluno ao lado que esta classe possui vários atributos (prontuario, nome, nota1 e nota2) e que estes atributos são privados!!!
- Os métodos Get e Set tem este nomes pois são métodos desta classe Aluno que permite alterar os atributos dos objetos (lembre que como os atributos são private uma variável referência para este objeto não acessa os atributos).
- Get: métodos para obter/retorna o valor do atributo.
- Set: método para atribuir/alterar o valor do atributo

Aluno
- prontuario: String - nome: String - nota1: double - nota2: double
// colocamos aqui os métodos get e set obterMedia():double mostrarDados(): void

Métodos Get e Set

- Os métodos Get e Set seguem um padrão de nome. Eles são seguido do nome do atributo que irão alterar (set) ou obter (get) com o nome do atributo iniciando em Maiúscula
- Para esta classe teremos os métodos:
 - Métodos Get
 - getProntuario()
 - getNome()
 - getNota1()
 - getNota2()
 - Método Set
 - setProntuario
 - setNome()
 - setNota1()
 - setNota2()

Aluno
<ul style="list-style-type: none">- prontuario: String- nome: String- nota1: double- nota2: double
<pre>// colocamos aqui os métodos get e set obterMedia():double mostrarDados(): void</pre>

Métodos Get e Set

- Os métodos Get obtém/retorna o valor do atributo então ele deve ter em sua representação no exemplo para o atributo prontuário e nota1 a seguinte notação (mostrada também na figura)
 - + getProntuario(): String // não tem parâmetro e retorna um tipo String
 - + getNota1(): double // não tem parâmetro e retorna um tipo double
- Os métodos Set altera/atribui um valor ao atributo então ele deve ter em sua representação no exemplo para o atributo prontuário e nota1 a seguinte notação (mostrada também na figura)
 - + setProntuario(String):void //recebe um parâmetro String e não retorna nada
 - + setNota1(double): void //recebe um parâmetro double e não retorna nada

O mesmo padrão segue para os outros atributos das classe não representado na figura!!!!

Aluno
<ul style="list-style-type: none">- prontuario: String- nome: String- nota1: double- nota2: double
<pre>// colocamos aqui os métodos get e set + setProntuario(String):void + setNota1(double): void + getProntuario(): String + getNota1(): double + obterMedia():double + mostrarDados(): void</pre>

Implementando os Métodos Get e Set em Java

```
// obtem o prontuario
String getProntuario() {
    return prontuario;
}

// obtem a nota1
double getNota1() {
    return nota1;
}
```

```
// altera prontuario do aluno
void setProntuario(String pront) {
    prontuario = pront;
}

// altera a nota1 do aluno
void setNota1(double n1) {
    nota1 = n1;
}
```

Aluno
<ul style="list-style-type: none">- prontuario: String- nome: String- nota1: double- nota2: double
<pre>// colocamos aqui os métodos get e set + setProntuario(String):void + setNota1(double): void + getProntuario(): String + getNota1(): double + obterMedia():double + mostrarDados(): void</pre>

O mesmo padrão segue para os outros atributos das classe !!!

Classe e Testado Aluno: atributos e métodos get e set

```
public class Aluno {  
    private String prontuario;  
    private String nome;  
    private double nota1;  
    private double nota2;  
  
    double obterMedia () {  
        média = ( nota1 + nota2 ) / 2;  
    }  
    void setNome(String no) {  
        nome = no;  
    }  
    String getNome(){  
        return nome;  
    }  
    void setProntuario(Sring pront) {  
        prontuario = pront;  
    }  
    void setProntuario(Sring pront) {  
        prontuario = pront;  
    }  
    // get e set do outros atributos ...
```

```
public class AlunoTestadora {  
    public static void main( String [ ] arts )  
    {  
        Aluno a1 = new Aluno(); // cria um objeto Aluno  
        Aluno a2 = new Aluno(); // cria outro objeto Aluno  
        a1.setNome("Jose" ); //a1 passa a ter nome Jose  
        a2.setNome("Aninha" ); //a2 passa a ter nome Aninha  
        System.out.printf("Nome do Aluno1:,%s", a1.getNome( ));  
        System.out.printf("Nome do Aluno2:,%s", a2.getNome( ));  
    }  
}
```

Veja com cuidado as chamadas dos métodos:

- operador . (ponto)
- chamador?
- argumentos
- se o método chamado tiver retorno?