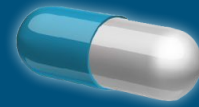


Encapsulamento



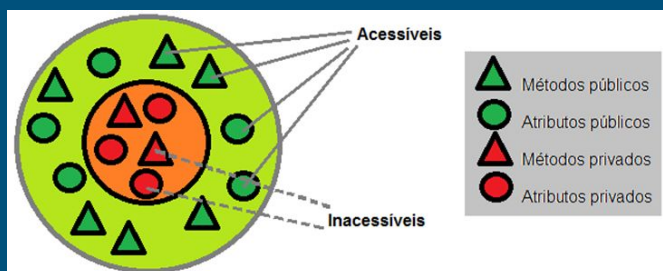
"Esconder / Proteger"

emerson@paduan.pro.br

Conceito

É necessário garantir que os dados (atributos) dos objetos não sejam modificados de forma indevida.

A modificação de forma indevida interfere no funcionamento do objeto e pode gerar erros.



emerson@paduan.pro.br

Modificadores

Modificador	Funcionalidade
public	permite que qualquer outra parte da aplicação tenha acesso ao membro
Padrão (default)	membros que não foram marcados com nenhum modificador explicitamente. Só podem ser acessados por outras classes dentro do mesmo pacote
protected	os membros são acessíveis por classes dentro do mesmo pacote e por classes derivadas (mesmo em pacotes diferente)
private	só é acessível dentro da própria classe em que foi declarado

emerson@paduan.pro.br

get / set

Em Java utilizamos nomes de métodos iniciando com get ou set para indicar métodos que alteram ou obtém valores de atributos privados.

Exemplo:

getHora: obtém o valor do atributo hora

setHora: altera o valor do atributo hora

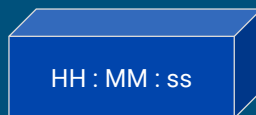
emerson@paduan.pro.br

Exercício 1 - Relógio

Escreva a classe Relógio, com os atributos hora, minuto e segundo, e com um construtor que recebe horas, minutos e segundos para inicializar o relógio. Faça um método da classe para exibir a hora atual. Crie os get's e set's para os atributos.

Faça um programa (main) que crie um objeto do tipo Relógio e exiba a hora atual do relógio.

O relógio não pode aceitar valores inválidos



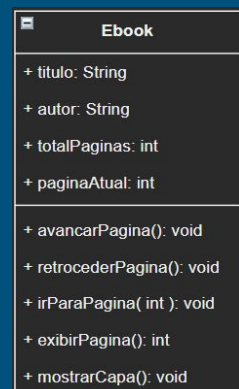
Relógio

emerson@paduan.pro.br

Exercício 2 - Ebook

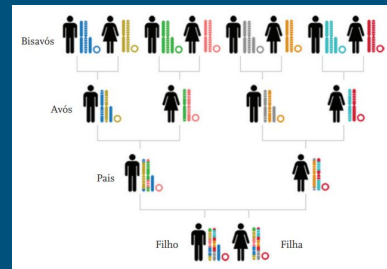
Considere que você está desenvolvendo um aplicativo de livro eletrônico. Os atributos e métodos estão no diagrama. As funcionalidades incluem exibir em que página está agora, para que página foi, etc.

Escreva uma classe para implementar estas funcionalidades e faça o app (main) para testar essa navegação entre as páginas.



emerson@paduan.pro.br

Herança



receber dos antepassados

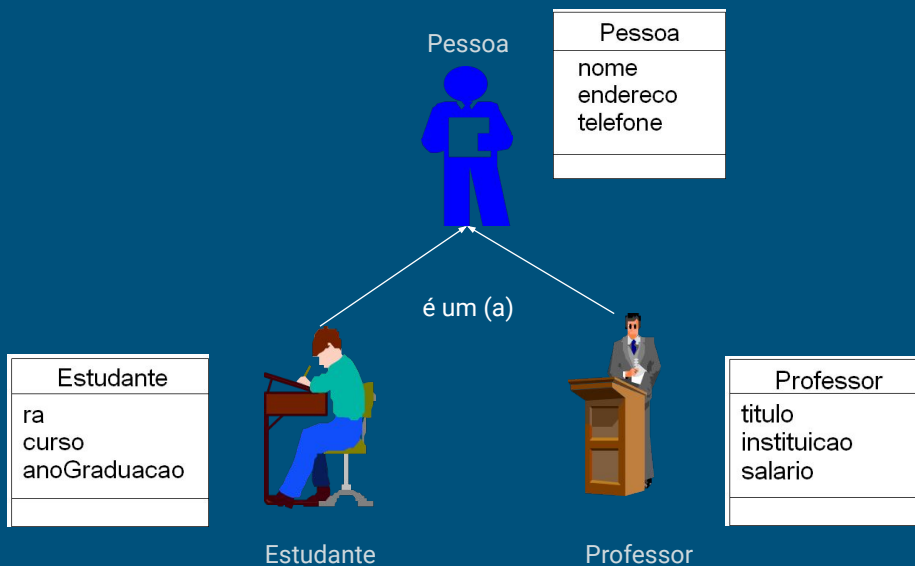
emerson@paduan.pro.br

Herança

Herança é um mecanismo que permite que características comuns a diversas classes sejam derivadas de uma classe base, ou superclasse.

A herança é uma forma de reutilização de software em que novas classes são criadas a partir das classes existentes, herdando seus atributos e métodos e adicionando novos recursos que as novas classes exigem.

emerson@paduan.pro.br



emerson@paduan.pro.br

Em Java

```
public class SuperClass {  
    //corpo da superclasse...  
}
```

extends - indica que a criação de uma nova classe que **herda** de uma classe existente

```
public class SubClass extends SuperClass {  
    //corpo da subclasse...  
}
```

emerson@paduan.pro.br

Exercício 3

