

Encapsulamento

Encapsulamento

Encapsulamento, vem de encapsular, que em programação orientada a objetos significa juntar o programa em partes, o mais isoladas possível. A ideia é tornar o software mais flexível, fácil de modificar e de criar novas implementações.

Percebemos nos exemplos anteriores utilizamos a palavra **public**, ou seja, conseguimos acessar esse membro da classe (esse membro não está ocultado) , mas poderia ser de outros dois tipos diferentes: **protected** ou **private** .

Diferenças entre essas 3 capacidades:

- **public** → é acessível de qualquer lugar fora da classe, mas dentro de um programa. Você pode definir e obter o valor de variáveis públicas sem nenhum membro;
- **protected** → não pode ser acessada ou mesmo visualizada fora da classe (classe derivada), mas pode ser acessada por classes derivadas.
- **private** → não pode ser acessada ou mesmo visualizada fora da classe (classe derivada). Somente os membros que estão dentro da mesma classe;



Encapsulamento

Quadro para melhor compreensão:

Base para comparação	public	private	protected
Herdando propriedade para a classe derivada	A classe derivada PODE acessar membros da classe base.	A classe derivada NÃO pode acessar membros privados da classe base.	Classe derivada pode acessar membros protegidos da classe base.
Acessibilidade	qualquer membro é acessível	Os membros privados da classe são inacessíveis fora do escopo da classe.	Os membros protegidos da classe são inacessíveis fora do escopo de classe, exceto a classe derivada imediatamente.
Acessível da própria class	sim	sim	sim
Acessível da classe derivada	sim	não	sim
Acessível de fora	sim	não	não