public, private e protected

ENCAPSULAMENTO

- Se outros programadores usam nossa classe, nós queremos garantir que erros pelo mau uso não ocorram. Encapsulamento serve para controlar o acesso aos atributos e métodos de uma classe.
- É uma forma eficiente de proteger os dados manipulados dentro da classe, além de determinar onde esta classe poderá ser manipulada.
- Usamos o nível de acesso mais restritivo que faça sentido para um membro particular. Sempre usamos private, a menos que tenhamos um bom motivo para deixá-lo com outro nível de acesso.

- Não devemos permitir o acesso público aos membros, exceto em caso de ser constantes. Isso porque membros públicos tendem a nos ligar a uma implementação em particular e limita a nossa flexibilidade em mudar o código.
- Basicamente, usamos quatro tipos de encapsulamento que são divididos em dois níveis:
- Nível de classe ou topo: Quando determinamos o acesso de uma classe inteira que pode ser public ou package-private (padrão);
- Nível de membro: Quando determinamos o acesso de atributos ou métodos de uma classe que podem ser public, private, protected ou package-private (padrão).

Para utilizarmos estes modificadores de acesso, basta que nós os digitemos antes do nome da variável, atributo, método, função ou classe, com exceção de package-private, que é entendido como padrão, portanto, é qualquer membro ou classe que não tenha modificador especificado.

- public class MinhaClasse { //classe public
- private int inteiro; //atributo inteiro private
- protected float decimal; //atributo float protected
- boolean ativado; //atributo booleano package-private

Public

 O modificador public deixará visível a classe ou membro para todas as outras classes, subclasses e pacotes do projeto Java.

Private

 O modificador private deixará visível o atributo apenas para a classe em que este atributo se encontra.

Protected

O modificador protected deixará visível o atributo para todas as outras classes e subclasses que pertencem ao mesmo pacote. A principal diferença é que apenas as classes do mesmo pacote tem acesso ao membro. O pacote da subclasse não tem acesso ao membro.

Sem Modificador (Padrão)

- Por padrão, a linguagem Java permite acesso aos membros apenas ao pacote em que ele se encontra.
- De forma ilustrativa, abaixo está uma tabela demonstrando todas estas características.

Modificador	Classe	Pacote	Subclasse	Globalmente
Public	sim	sim	sim	sim
Protected	sim	sim	sim	não
Sem Modificador (Padrão)	sim	sim	não	não
Private	sim	não	não	não
Fonte: Controlling Access to Members of a Class				