### Modificador de Acesso protected

O modificador de acesso protected em Java concede um nível de visibilidade intermediário entre private e public. Ele permite que:

**A classe que declara o membro:** Tenha acesso direto ao membro.

**As subclasses:** Tenham acesso ao membro, mesmo que estejam em um pacote diferente.

**As classes do mesmo pacote:** Tenham acesso ao membro.

**No contexto de herança:** O protected é crucial para permitir que as subclasses extendam e reutilize o comportamento da classe pai, sem expor todos os detalhes internos. Isso promove o encapsulamento e a reutilização de código.

2 - A palavra-chave super em Java é utilizada para:

**Referenciar membros da classe pai:** Quando uma classe filha possui um membro com o mesmo nome de um membro da classe pai, super permite acessar o membro da classe pai explicitamente.

**Chamar construtores da classe pai:** Ao criar um objeto de uma classe filha, o construtor da classe pai é chamado automaticamente. Se você precisar passar argumentos específicos para o construtor da classe pai, você pode usar super para isso.

3 - A herança funciona de forma similar para atributos e métodos, mas existem algumas nuances:

* **Atributos:** Quando uma classe filha herda de uma classe pai, ela herda todos os atributos não privados da classe pai. No entanto, a classe filha pode sobrescrever os valores desses atributos.
* **Métodos:** A classe filha pode sobrescrever os métodos da classe pai, fornecendo uma implementação diferente. A palavra-chave @Override é usada para indicar explicitamente que um método está sendo sobrescrito.

4 - **public:** Acessível de qualquer lugar do programa.

**private:** Acessível apenas dentro da classe onde é declarado.

**protected:** Acessível dentro da classe, em subclasses e em classes do mesmo pacote.