**En este caso, el atributo nombre está declarado como private, lo que significa que solo puede ser accedido dentro de la propia clase Persona. En cambio, el atributo edad tiene acceso de paquete (default), lo que permite que cualquier clase dentro del mismo paquete pueda acceder a él directamente.**

**Si nos fijamos en la clase de prueba Main, como está en el mismo paquete que Persona, puede acceder sin problemas al atributo edad. Sin embargo, no puede hacer lo mismo con nombre, ya que es privado. Para leer o modificar nombre, es necesario utilizar los métodos getNombre() y setNombre(), conocidos como getters y setters. Estos métodos permiten acceder y modificar los atributos privados de una manera controlada, lo que refuerza la encapsulación.**

### **Discusión sobre private y el acceso de paquete (default)**

* **private: Es el modificador de acceso más restrictivo. Cualquier atributo o método declarado como private solo es accesible dentro de la propia clase. Esto proporciona un alto nivel de encapsulación y protección de datos, ya que evita que otras clases accedan o modifiquen directamente los atributos internos.**
* **Acceso de paquete (default): Si no especificamos un modificador de acceso, el miembro será accesible desde cualquier clase dentro del mismo paquete. Aunque es menos restrictivo que private, sigue ofreciendo cierto grado de encapsulación al impedir el acceso desde clases de otros paquetes.**

**En definitiva, la elección entre private y el acceso de paquete depende de los objetivos de diseño del programa. Si buscamos una encapsulación fuerte y proteger los datos de modificaciones externas, lo mejor es usar private. En cambio, si queremos que ciertas clases dentro del mismo paquete puedan acceder a los atributos sin necesidad de métodos adicionales, el acceso de paquete puede ser una opción válida.**

**¡Espero que esta explicación ayude a entender mejor la diferencia entre estos modificadores de acceso!**