

Un `ArrayList` en Java es una implementación de la interfaz `List` que proporciona una estructura de datos de lista dinámica, similar a los arreglos, pero con la capacidad de redimensionarse automáticamente cuando se añaden o eliminan elementos. A diferencia de los arreglos, los `ArrayList` pueden cambiar de tamaño dinámicamente.

El paradigma de programación orientada a objetos (OOP) es un enfoque de diseño y desarrollo de software que organiza el código en torno a objetos y clases. Este paradigma se basa en cuatro pilares fundamentales: encapsulamiento, abstracción, herencia y polimorfismo. A continuación, se describen estos pilares en detalle:

## 1. Encapsulamiento

**Definición:** El encapsulamiento es el mecanismo de restringir el acceso directo a algunos de los componentes de un objeto. Esto se logra mediante el uso de modificadores de acceso (como `private`, `protected`, y `public` en Java).

**Definición:** La abstracción es el proceso de simplificar la complejidad al ocultar los detalles innecesarios para resaltar la información relevante. En OOP, esto se logra mediante la creación de clases abstractas e interfaces.

**Propósito:**

- Facilitar el manejo de la complejidad del sistema.
- Proporcionar una vista simplificada y específica del comportamiento de los objetos.

**Definición:** La herencia es el mecanismo por el cual una clase (subclase) puede heredar propiedades y comportamientos (métodos) de otra clase (superclase). Esto permite la reutilización del código y la creación de jerarquías de clases.

**Propósito:**

- Reutilizar el código existente.
- Establecer una relación de jerarquía entre las clases.
- Promover la consistencia y la extensibilidad del sistema.

**Definición:** El polimorfismo permite a los objetos de diferentes clases ser tratados como objetos de una clase común. El comportamiento específico de un método se determina en tiempo de ejecución.

**Propósito:**

- Permitir el uso de una interfaz común para diferentes tipos de objetos.
- Facilitar la extensibilidad y la flexibilidad del código.
- Promover la reutilización del código.