**UNIVERSIDAD DE LAS FUERZAS ARMADAS “ESPE”**

**NOMBRE:** ERICK DAMIAN ARIAS SARANGO

**MATERIA:** PROGRAMACION ORIENTADA A OBJETOS

**NRC :**16362

1. **RESPONDER LAS SIGUIENTES PREGUNTAS**

**¿Qué significa alta cohesión y bajo acoplamiento?**

**Alta cohesión:** Se refiere a la medida en que los elementos dentro de un módulo o componente de software están estrechamente relacionados entre sí. En otras palabras, un módulo con alta cohesión tiene componentes que están fuertemente relacionados y trabajan juntos para cumplir una función específica. Esto significa que un módulo hace una cosa y la hace bien. La alta cohesión facilita la comprensión del código, su mantenimiento y su reutilización.

**Bajo acoplamiento:** Se refiere a la medida en que los módulos o componentes de software están interconectados entre sí. Un bajo acoplamiento implica que los módulos son independientes entre sí y no están fuertemente vinculados. Esto significa que los cambios en un módulo tienen un impacto mínimo en otros módulos. Un bajo acoplamiento permite una mayor modularidad, flexibilidad y facilidad para realizar cambios en el código sin afectar otras partes del sistema.

**¿Qué son las clases wrapper?**

Las "clases wrapper" son clases que se utilizan para envolver o encapsular tipos de datos primitivos (como enteros, flotantes, caracteres, etc.) en objetos. Esto se hace para permitir que estos tipos de datos primitivos sean tratados como objetos dentro del paradigma de la programación orientada a objetos.

Por ejemplo, en Java, existen clases wrapper para tipos de datos primitivos:

**Integer** para envolver enteros.

**Float** para envolver flotantes.

**Character** para envolver caracteres, entre otros.

Estas clases wrapper proporcionan métodos y funcionalidades adicionales que no están disponibles en los tipos de datos primitivos, como métodos para convertir entre tipos de datos, comparar objetos, realizar operaciones matemáticas, entre otros.

El uso de clases wrapper es útil en situaciones donde se necesita trabajar con tipos de datos primitivos como objetos, por ejemplo, al utilizar colecciones (como Listas o Mapas) que solo pueden contener objetos en lugar de tipos de datos primitivos.