Investigación

Que son los polimorfismos

- Cuando se habla de polimorfismos en programación nos referimos a que una clase de nuestro código pueda sobrescribir a su clase base sin que esto afecte al funcionamiento de nuestro código.

Ejemplo polimorfismo en ejecución

-
```
class Animal {
    void sonido() {
        System.out.println("El animal hace un sonido");
    }
}
class Perro extends Animal {
    @Override
    void sonido() {
        System.out.println("El perro ladra");
    }
}
Animal miAnimal = new Perro();
miAnimal.sonido();  // Salida: El perro ladra
```

Principios y prácticas del código limpio

1. Desacoplamiento
    a. Minimizar las dependencias entre componentes del código.
    b. Utilizar interfaces y abstracciones para reducir el acoplamiento.
2. Módulos Bien Definidos
    a. Organizar el código en módulos claros y coherentes.
    b. Cada módulo debe tener una única responsabilidad.
3. Nombres Consistentes y Con Significado
    a. Usar nombres de variables, funciones y clases que transmitan claramente su propósito.
    b. Seguir una convención de nomenclatura coherente en todo el código.
4. Comentarios Útiles
    a. Escribir comentarios que expliquen el porqué del código.
    b. Evitar comentarios que solo describan lo que hace el código.
5. Evitar Código Duplicado
    a. Reutilizar funciones y clases para evitar duplicación.
    b. Refactorizar código duplicado en métodos comunes.
6. Manejo de Errores y Excepciones
    a. Manejar errores de manera consistente y clara.

    b. Usar excepciones en lugar de códigos de error.
7. Principio de Responsabilidad Única (SRP)
    a. Cada clase debe tener una sola responsabilidad.
8. Principio de Segregación de la Interfaz (ISP)
    a. Preferir muchas interfaces pequeñas y específicas en lugar de una sola interfaz grande.
9. Pruebas Unitarias
    a. Escribir pruebas unitarias para asegurar que el código funcione como se espera.
10. Refactorización Continua
    a. Mejorar constantemente el código sin cambiar su funcionalidad.
11. Principio de Sustitución de Liskov (LSP)
    a. Las clases derivadas deben poder sustituir a sus clases base sin alterar el comportamiento del programa.

Bibliografía:

- Los 25 Principios Básicos de la Programación Orientada a Objetos
- SOLID: qué es y cuáles son los 5 principios de la Programación Orientada a Objetos (POO) | Alura Cursos Online
- Página no encontrada - Adictos al trabajo
- Polymorphism (The Java™ Tutorials > Learning the Java Language > Interfaces and Inheritance)
- cplusplus.com/doc/tutorial/polymorphism/