Universidad de La Laguna

Patrón estrategia

DAA

Darío Domínguez González……..……..……alu0101408095@ull.edu.es

El patrón de diseño **Estrategia (Strategy)** es un patrón de comportamiento que permite definir una familia de algoritmos, encapsular cada uno en una clase independiente y hacer que sean intercambiables. Su objetivo principal es permitir que un objeto pueda cambiar su comportamiento en tiempo de ejecución sin modificar su código fuente. Esto se logra mediante la delegación de la implementación del comportamiento a diferentes clases de estrategia.

Componentes principales del patrón Estrategia:

1. **Contexto:** Es la clase que contiene una referencia a una estrategia y delega en ella la ejecución del comportamiento específico. El contexto no necesita conocer los detalles internos de cada estrategia, solo la interfaz común que implementan.
2. **Estrategia (Interfaz):** Define un contrato común para todas las estrategias, garantizando que el contexto pueda interactuar con ellas de manera uniforme. Esto permite que las estrategias sean fácilmente intercambiables.
3. **Estrategias Concretas:** Son las clases que implementan la interfaz de estrategia y proporcionan distintas implementaciones del comportamiento deseado. Cada una representa una forma específica de realizar una tarea determinada.

Beneficios de utilizar el patrón Estrategia:

- **Flexibilidad:** Permite cambiar dinámicamente el comportamiento de un objeto sin modificar su código. Esto es útil cuando un sistema necesita soportar múltiples variaciones de un mismo proceso.
- **Mantenimiento y escalabilidad:** Facilita la adición o modificación de algoritmos sin alterar el código del contexto. Esto sigue el **principio de abierto/cerrado**, que sugiere que las entidades de software deben estar abiertas para su extensión, pero cerradas para su modificación.
- **Reutilización de código:** Al separar los algoritmos en estrategias independientes, estos pueden ser reutilizados en diferentes contextos o aplicaciones sin necesidad de duplicar código.
- **Reducción del acoplamiento:** El contexto y las estrategias concretas están desacopladas, lo que permite desarrollar cada una de manera independiente y hacer pruebas unitarias más efectivas.

En resumen, el patrón Estrategia es una herramienta poderosa en el diseño de software, ya que mejora la organización del código, facilita su mantenimiento y permite una mayor adaptabilidad a cambios futuros sin comprometer la estructura del sistema.