Why Bridge is a Structural Pattern and Not a Behavioral Pattern

This document explains why the Bridge Design Pattern is classified as a Structural Pattern rather than a Behavioral Pattern, despite involving communication between objects.

Behavioral Patterns vs. Structural Patterns:

1. Behavioral Patterns:

- Focus: These patterns focus on how objects interact with each other, manage responsibilities, and communicate. The emphasis is on the flow of control and the delegation of behavior among objects.

- Examples: Observer, Strategy, Command, and Mediator patterns.

- Key Concern: The key concern is how different objects collaborate to achieve a specific behavior or complete a task. The patterns often involve defining protocols for object interactions.

2. Structural Patterns:

- Focus: These patterns focus on how objects and classes are composed to form larger structures. The emphasis is on organizing classes and objects into larger structures while ensuring that the system remains flexible and efficient.

- Examples: Adapter, Bridge, Composite, and Decorator patterns.

- Key Concern: The key concern is how to compose objects and classes to create structures that are easy to manage, extend, and modify. Structural patterns deal with the organization of object hierarchies and the relationships between classes.

Why Bridge is a Structural Pattern and Not a Behavioral Pattern:

- Decoupling of Abstraction and Implementation:

  - The Bridge Pattern is primarily concerned with decoupling. It decouples the abstraction (e.g., RemoteControl) from its implementation (e.g., Device). The goal is to allow both the abstraction and implementation to vary independently without affecting each other. This decoupling is a hallmark of structural design patterns because it changes how the system is organized.

- Composition Over Interaction:

  - In the Bridge Pattern, the interaction between objects (e.g., between RemoteControl and Device) is not about defining how they communicate to achieve a particular behavior. Instead, it's about composing these objects in such a way that the structure of the system remains flexible and easy to extend.
  - The pattern does not define the communication protocol between objects; it defines how to organize and separate the concerns of abstraction and implementation.

- Focus on Structure:

  - The primary focus of the Bridge Pattern is on the structure of the codebase. It addresses the problem of creating a flexible and maintainable codebase by separating the abstraction from the implementation. This separation allows new devices or remote controls to be added without changing the existing structure, which is a key concern in structural patterns.

Comparison to Behavioral Patterns:

- In Behavioral Patterns, like the Observer Pattern, the main concern is how an object informs other objects about changes in its state. The focus is on the protocols of communication and the flow of information.
- In Bridge, the interaction between RemoteControl and Device exists, but it is secondary to the primary concern of how to structure the code to make it flexible and extensible. The communication in Bridge is more about invoking operations on another class, rather than defining how objects should behave or react to one another.

Summary:

The Bridge Pattern is a structural pattern because it is primarily concerned with how the classes and objects are organized in a system to create a flexible and scalable architecture. While there is communication between objects (which happens in almost every design pattern), the key focus is on decoupling the abstraction from the implementation, which is a structural concern. The interactions between objects in Bridge are about managing structure, not about defining behavior or communication protocols.