Structural design patterns

**Structural design patterns** are concerned with how classes and objects can be composed, to form larger structures.

The structural design patterns **simplifies the structure by identifying the relationships**.

These patterns focus on, how the classes inherit from each other and how they are composed from other classes.

Types of structural design patterns

There are following 7 types of structural design patterns.

1. [Adapter Pattern](https://www.javatpoint.com/adapter-pattern)

Adapting an interface into another according to client expectation.

1. [Bridge Pattern](https://www.javatpoint.com/bridge-pattern)

Separating abstraction (interface) from implementation.

1. [Composite Pattern](https://www.javatpoint.com/composite-pattern)

Allowing clients to operate on hierarchy of objects.

1. [Decorator Pattern](https://www.javatpoint.com/decorator-pattern)

Adding functionality to an object dynamically.

1. [Facade Pattern](https://www.javatpoint.com/facade-pattern)

Providing an interface to a set of interfaces.

1. [Flyweight Pattern](https://www.javatpoint.com/flyweight-pattern)

Reusing an object by sharing it.

1. [proxy Pattern](https://www.javatpoint.com/proxy-pattern)

Representing another object.