

# Structural pattern

From Wikipedia, the free encyclopedia

In software engineering, **structural design patterns** are design patterns that ease the design by identifying a simple way to realize relationships between entities.

Examples of Structural Patterns include:

- Adapter pattern: 'adapts' one interface for a class into one that a client expects
  - Adapter pipeline: Use multiple adapters for debugging purposes.<sup>[1]</sup>
  - Retrofit Interface Pattern:<sup>[2][3]</sup> An adapter used as a new interface for multiple classes at the same time.
- Aggregate pattern: a version of the Composite pattern with methods for aggregation of children
- Bridge pattern: decouple an abstraction from its implementation so that the two can vary independently
  - Tombstone: An intermediate "lookup" object contains the real location of an object.<sup>[4]</sup>
- Composite pattern: a tree structure of objects where every object has the same interface
- Decorator pattern: add additional functionality to a class at runtime where subclassing would result in an exponential rise of new classes
- Extensibility pattern: aka. Framework - hide complex code behind a simple interface
- Facade pattern: create a simplified interface of an existing interface to ease usage for common tasks
- Flyweight pattern: a large quantity of objects share a common properties object to save space
- Marker pattern: an empty interface to associate metadata with a class.
- Pipes and filters: a chain of processes where the output of each process is the input of the next
- Opaque pointer: a pointer to an undeclared or private type, to hide implementation details
- Proxy pattern: a class functioning as an interface to another thing

## See also

- Behavioral pattern
- Concurrency pattern
- Creational pattern

## References

1. "Adapter Pipeline". Cunningham & Cunningham, Inc. 2010-12-31. Archived from the original on 2010-12-31. Retrieved 2012-07-20.
2. BobbyWoolf (2002-06-19). "Retrofit Interface Pattern". Cunningham & Cunningham, Inc. Archived from the original on 2002-06-19. Retrieved 2012-07-20.
3. MartinZarate (2010-12-31). "External Polymorphism". Cunningham & Cunningham, Inc. Archived from the original on 2010-12-31. Retrieved 2012-07-20.
4. "Tomb Stone". Cunningham & Cunningham, Inc. 2007-06-17. Archived from the original on 2007-06-17. Retrieved 2012-07-20.



Wikimedia Commons has media related to ***Structural software design patterns***.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Structural\_pattern&oldid=727570611"

Categories: Software design patterns

- 
- This page was last modified on 29 June 2016, at 21:02.
  - Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.