Behavioral Patterns are concerned with providing solutions regarding object interaction - how they communicate, how are some dependent on others, and how to segregate them to be both dependent and independent and provide both flexibility and testing capabilities.

* [Interpreter](https://stackabuse.com/behavioral-design-patterns-in-java/#interpreter)
* [Template Method/Pattern](https://stackabuse.com/behavioral-design-patterns-in-java/#templatemethod)
* [Chain of Responsibility](https://stackabuse.com/behavioral-design-patterns-in-java/#chainofresponsibility)
* [Command](https://stackabuse.com/behavioral-design-patterns-in-java/#command)
* [Iterator](https://stackabuse.com/behavioral-design-patterns-in-java/#iterator)
* [Observer](https://stackabuse.com/behavioral-design-patterns-in-java/#observer)
* [State](https://stackabuse.com/behavioral-design-patterns-in-java/#state)
* [Strategy](https://stackabuse.com/behavioral-design-patterns-in-java/#strategy)
* [Visitor](https://stackabuse.com/behavioral-design-patterns-in-java/#visitor)