The **State design pattern** allows an object to change its behavior when its internal state changes. [It achieves this by encapsulating the object’s behavior within different state objects, and the object itself dynamically switches between these state objects depending on its current state1](https://www.geeksforgeeks.org/state-design-pattern/). Here are **five free resources** where you can learn more about the State design pattern:

1. [**GeeksforGeeks**: Provides an in-depth explanation of the State pattern along with examples and UML diagrams1](https://www.geeksforgeeks.org/state-design-pattern/).
2. [**Visual Paradigm Tutorials**: Offers a tutorial specifically focused on the State pattern, including practical examples and UML diagrams](https://www.geeksforgeeks.org/state-design-pattern/)[2](https://tutorials.visual-paradigm.com/state-pattern-tutorial/).
3. [**MethodPoet**: Discusses real-world examples and use cases for the State pattern, making it easier to understand and apply](https://www.geeksforgeeks.org/state-design-pattern/)[3](https://methodpoet.com/state-pattern/).
4. [**SourceMaking**: Explores the implementation details of the State pattern, including its relationship with the Strategy pattern](https://www.geeksforgeeks.org/state-design-pattern/)[4](https://sourcemaking.com/design_patterns/state).
5. [**Baeldung**: Provides a Java-based tutorial with UML diagrams and a practical example for understanding the State pattern](https://www.geeksforgeeks.org/state-design-pattern/)[5](https://www.baeldung.com/java-state-design-pattern).

Feel free to explore these resources to deepen your understanding of the State design pattern! 🚀