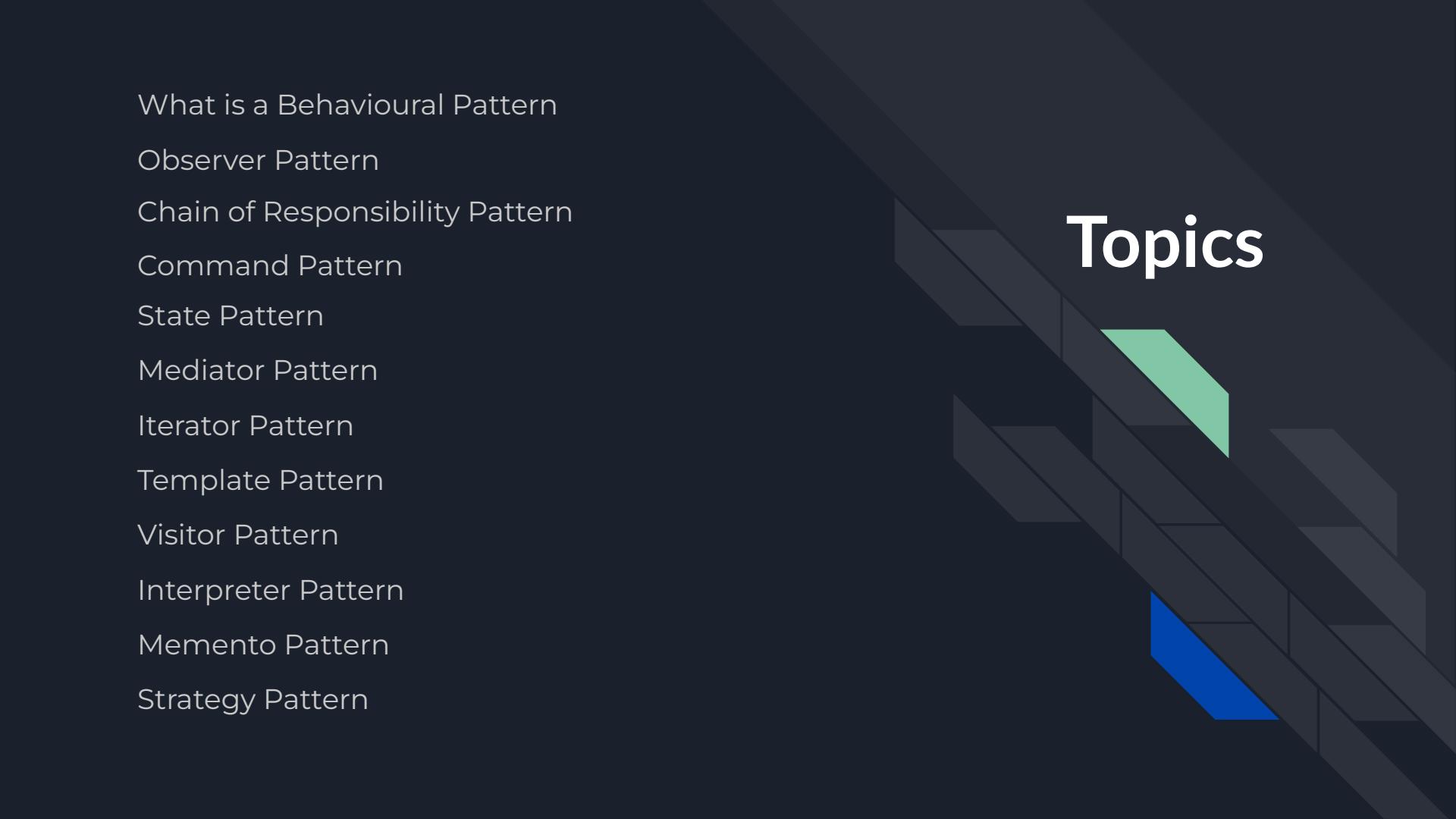




Behavioural Design Patterns

Presented by Antony Jude Shaman A

Topics



What is a Behavioural Pattern

Observer Pattern

Chain of Responsibility Pattern

Command Pattern

State Pattern

Mediator Pattern

Iterator Pattern

Template Pattern

Visitor Pattern

Interpreter Pattern

Memento Pattern

Strategy Pattern



What is a Behavioural Pattern

Behavioural pattern focus on:

- Communication between the objects
- How they interact and
- Responsibilities that are distributed among them.



Observer Pattern

- Observer pattern acts like an event listener that helps us to perform tasks when the state of an object changes.
- It helps in maintaining track of the state of the application



Chain of Responsibility Pattern

- It helps multiple handlers to process a request without the end-user knowing which handler is handling the request.
- Requests can be handled and sent to the handlers suitable for the request handler dynamically.



Command Pattern

- Command Pattern helps in encapsulation of requests as objects.
 - Invoker
 - Receiver
- The user does not need to know the operations performed to execute the command request.



State Pattern

- State pattern allows an object to change its behaviour when the internal state of the class changes.
- Behavior is defined by different state classes, each encapsulating specific actions for that state.



Mediator Pattern

- Mediator pattern helps in maintaining a centralized system to establish and maintain communication between the objects.
- The objects interact with each other through a mediator object



Iterator Pattern

- Iterator pattern lets us to traverse through a list, tree, linked list or graph without exposing the data structure used.
- It is useful to create a custom loop that has no knowledge and does not reveal the structure of our data.



Template Pattern

- Template pattern defines the skeleton of an algorithm in a superclass, allowing subclasses to define specific steps.
- The pattern promotes code reuse and consistency across similar processes while allowing customization for individual cases.



Visitor Pattern

- Visitor Pattern helps performing various kinds of operations by visiting classes and not modifying it.
- By using the visitor, you can add new operations or functionalities to the objects without modifying their existing code.



Interpreter Pattern

- Interpreter pattern defines a representation for its grammar along with an interpreter that uses the representation to interpret sentences in the language.
- The interpreter pattern is not suitable and efficient for large complex grammars or expressions.



Memento Pattern

- Memento pattern allows us to store the internal state of the system and restore to previous saved states.
- It has 3 components:
 - Originator
 - Memento
 - Caretaker



Strategy Pattern

- Strategy pattern lets us use different strategies(algorithms) for different scenarios.
- It helps in maintaining a family of algorithms and use only the required algorithm based on conditions



Thank you!

