Page No.:
Date:
YOUVA

Q. Understand what is design pattern in java.

→ A design pattern is a reusable solution for common problems in software design used in engineering. It is not a full design ready for coding but rather a guideline or model for solving issues. Design patterns can be adapted to different situations and contexts, providing flexibility in problem-solving issues. Design patterns can be adapted to different situations and context, providing flexibility in problem solving.

Q what purpose does it solve in design pattern?

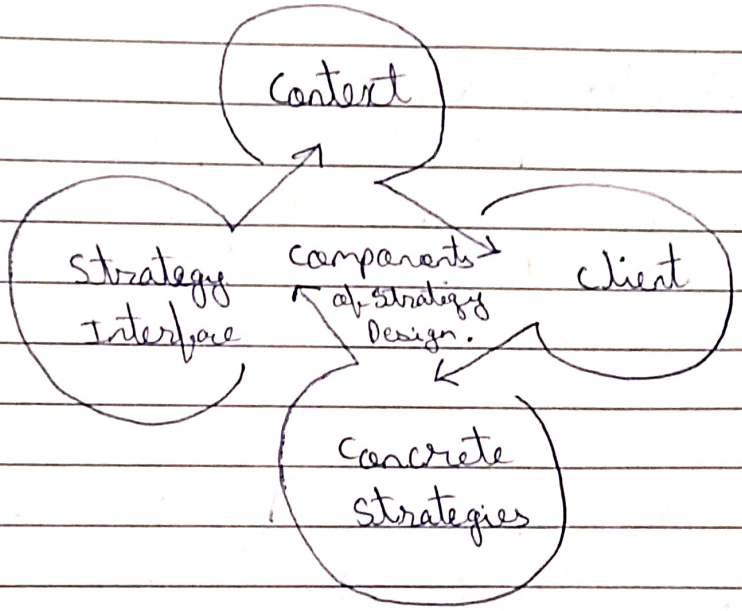
→ Use design patterns for common design problem with established solutions, providing proven strategies for software challenges.

→ Enhance code reusability, flexibility, and maintainability, making modifications easier as requirements evolve.

→ Support key principles like separation of concerns encapsulation, and dependency inversion, leading to better modularity and reduced dependencies.

Q. Strategy Design Pattern :-

The Strategy Design Patterns defines a family of algorithm, encapsulates each one, and makes them interchangeable, allowing clients to switch algorithms dynamically ~~not~~ without altering the code structure.



1. Context :-

A class or object known as context assigns the task to a strategy object and contains a reference to it.

- It serves as an intermediary between the client and the strategy, offering an integrated approach for task execution without exposing every detail of the process.

2) Strategy interface :-

It is an interface or abstract class that defines a set of methods that all concrete strategies must implement.

3) Concrete strategies :-

Concrete strategies are the various implementations of the strategy interfaces. Each concrete strategy provides a specific algorithm or behaviour for performing the task defined by the strategy interface.

4) Client :-

The client is responsible for selecting and configuring the appropriate strategy and providing it to the context.

Q. 5 different types of design pattern.

1) Chain of Responsibility method design pattern :-

Chain pattern is used to achieve loose coupling in software design where a request from the client is passed to a chain of objects to process them.

2) Command Method Design pattern :-

A behavioural design pattern called the command pattern transforms a request into an independent object with all of the information request

3) Interpreter method design pattern :-

Interpreter pattern is used to defines a grammatical representation for a language and provides and interpreter to deal with this grammar.

4) State method design pattern :-

It is possible to select an objects behaviour at runtime by utilizing the strategy Design Pattern.

Encapsulating a family of algorithms into distinct classes that each implement a common interface is the foundation of strategy pattern.

5) State method design pattern :-

When an object modifies its behaviour according to its internal state, the state design pattern is applied.
If we have to change the behaviour of an object based on its state, we can have a state variable in the object and use the if-else condition block to perform different actions based on the state.

5). Observer Method Design Pattern :-

It establishes a one-to-many dependency between objects, meaning that all of the dependencies (observers) of the subject are immediately updated and notified when the subject changes.