Command Pattern

A Command Pattern says that "*encapsulate a request under an object as a command and pass it to invoker object. Invoker object looks for the appropriate object which can handle this command and pass the command to the corresponding object and that object executes the command*".

It is also known as **Action or Transaction.**

Advantage of command pattern

* It separates the object that invokes the operation from the object that actually performs the operation.
* It makes easy to add new commands, because existing classes remain unchanged.

Usage of command pattern:

It is used:

* When you need parameterize objects according to an action perform.
* When you need to create and execute requests at different times.
* When you need to support rollback, logging or transaction functionality.

Example of command pattern

Let's understand the example of adapter design pattern by the above UML diagram.

UML for command pattern:

These are the following participants of the Command Design pattern:

* **Command** This is an interface for executing an operation.
* **ConcreteCommand** This class extends the Command interface and implements the execute method. This class creates a binding between the action and the receiver.
* **Client** This class creates the ConcreteCommand class and associates it with the receiver.
* **Invoker** This class asks the command to carry out the request.
* **Receiver** This class knows to perform the operation.

