The Strategy pattern is a behavioral design pattern that allows a set of behaviors to be turned into objects and made interchangeable inside an original context object.

This pattern is used when there are multiple algorithms for a specific task, and the client decides the actual implementation to be used at runtime.

First, we create a Strategy interface defining an action and concrete strategy classes implementing the Strategy interface.

Next, we create a Context class that uses a Strategy. The Context class has a reference to a Strategy object, and delegates executing the behavior to the linked strategy object. The Context class can change the way it performs its work by replacing the currently linked strategy object with another one.

The Strategy pattern is commonly used especially in various frameworks to provide users a way to change the behavior of a class without extending it.