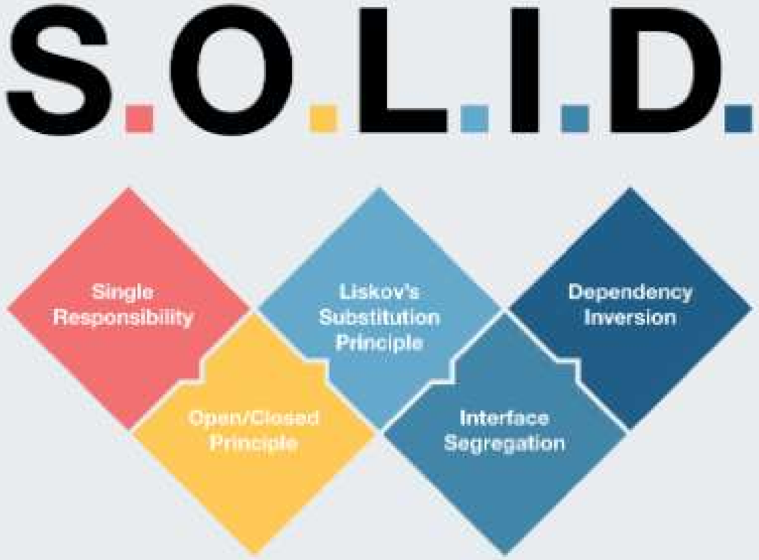
Single Responsibility Principle



Part 2

Visit GitHub @BCAPATHSHALA

1. What is single Responsibility Principle (SRP)

1. A class should have one, and only one reason to change.
   This means that a class should only have one job or responsibility.
2. A class should only be responsible for one thing.
3. There's a place for everything and everything in its place.
4. Find one reason to change and take everything else out of the class.
5. Importance: Following **`SRP`** makes your code more modular, easier to understand, maintain, and extend.
   It helps in isolating functionalities, making debugging and testing more straightforward.

In One Statement

The Single Responsibility Principle states that a class should have only one reason to change, meaning it should have only one job or responsibility. This promotes modularization and makes the code easier to understand and maintain.

Key Idea:

A class should do only one thing, and it should do it well.

Real-Time Examples:

1. Think of a chef who only focuses on cooking, not managing the restaurant or delivering food.
2. In auth system, user only focuses on login, not sending email or verify email
3. In a household setting, the book container exclusively stores books, the food bucket is designated for food items only, and the tool box is specifically reserved for tools, ensuring each container serves its intended purpose without mixing items.

How can Single Responsibility Principle be applied?

Visit GitHub:
@BCAPATHSHALA

Practical Coding Examples in Java #1
Practical Coding Examples in Java #2
Practical Coding Examples in Java #3