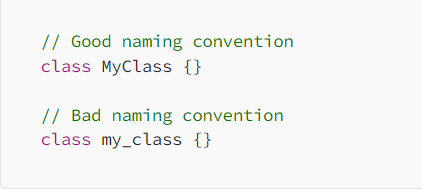
Clean code helps to ensure that your code is understandable, maintainable, and readable.

Flutter Clean Code Practices:

1. Follow the Dart naming conventions (Pascal Case/ Upper Camel Case)

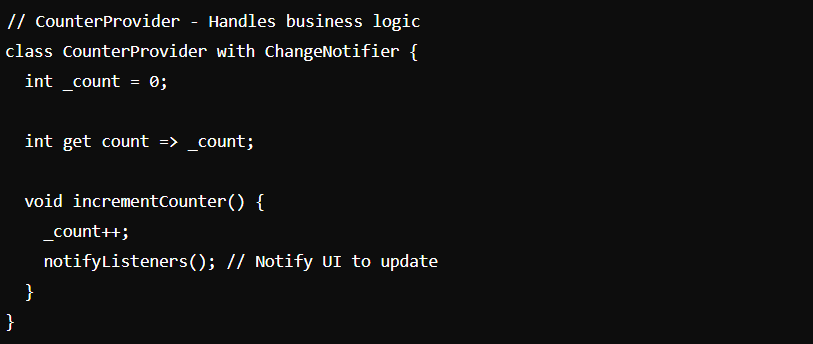


1. Descriptive Variable and Function names
2. Proper indentation and formatting
3. Single Responsibility Principle (SRP): A single class/function should be responsible for a single functionality only and should not be responsible for multiple tasks. For example separating UI components, business logic, and data access logic into separate classes.

A screenshot of a computer program

Description automatically generated In this example, it may seem that SRP overcomplicates code but this is a simple example. SRP makes scaling easier and is also helpful when you have multiple screens/components that need to access the same business logic or modify the same data. It’s also helpful when you need to change where your data is coming from or add business logic without messing up the other parts.

In flutter state management solutions like provider and bloc can be used to enforce SRP.

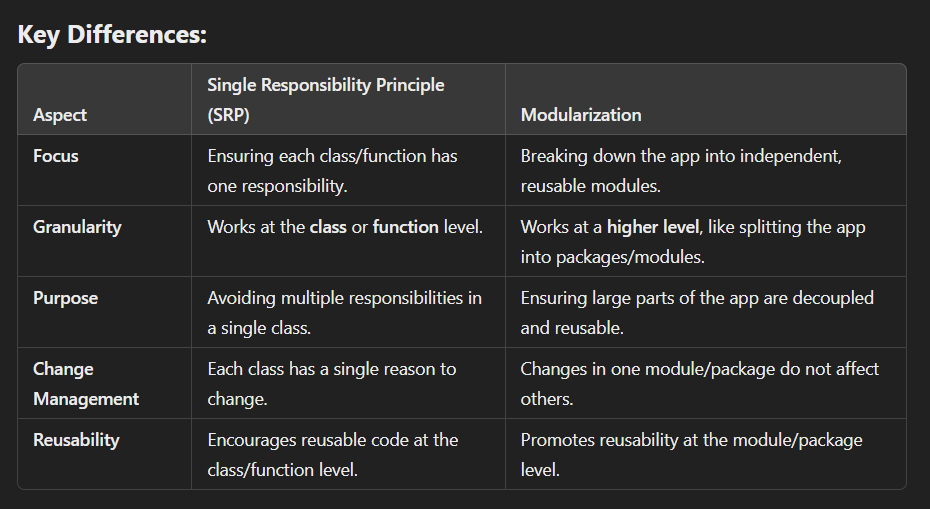
For example provider can be used to separate business logic like so: A screenshot of a computer program

Description automatically generatedA computer screen with white text

Description automatically generated

1. Modularization: Your entire application should be made up of smaller self-contained modules. Each module typically encapsulates a certain feature or aspect of the application and is independently testable and maintainable.

SRP focuses on each class/function having a single responsibility whereas modularity focuses on breaking down your app into simpler, independent components (splitting your app into packages or modules).



1. Test Driven Development: writing tests for your code before writing the code itself to ensure that code behaves as expected and catch errors early in the development process.
2. Use the proper flutter architecture: The Flutter framework supports several architectures, such as MVC, MVP, MVVM, and Clean Architecture. Choose the right architecture for your project based on its complexity and requirements. Use the Flutter Bloc library to implement Clean Architecture in your project.