

Encapsulation

SEALED (CLASS)

A `sealed class` cannot be inherited (no other class can extend it).

READONLY (VARIABLE)

A `readonly variable` can only be assigned a value at the time it is declared. it can't be changed after that.

PRIVATE

A `private variable or method` can only be accessed within the class itself, not from outside.

PROTECTED

A `protected variable or method` can be accessed inside the class and also by any class that `inherits` from it.

PUBLIC

A `public variable or method` can be accessed from anywhere.

ENUM

An `enum` is a way to define a list of related constants with readable names.

WHAT IS ENCAPSULATION?

`Encapsulation` is a key concept in object-oriented programming (OOP) where we bundle the data (variables) and the methods (functions) that operate on the data into a single unit, called a **class**. It also involves hiding the internal details of how the data is managed, allowing access only through controlled methods.

WHY DO WE USE ENCAPSULATION?

- To protect data from being changed directly.
- To make code more organized and secure.
- To control how data is accessed or modified.

HOW DO WE USE ENCAPSULATION IN OUR CODE?

- Access Modifiers like `private`, `public` and `protected` to control what is accessible from outside the class.