Flutter Required Task3

التاريخ: 9/12/24

الموضوع:

1. OOP in Flutter

First, what is OOP?
OOP is a programming paradigm allows us to organize our code style and make it reflects the real world as possible.

⊛ In OOP, any entity will be represented as a class that has all properties associated to this entity and all behaviours it should have. this class is a template and becomes useful when we take an object From it. because our program is about objects interact with each others.
In Dart we use class keyword to define a class.
And use pascal Notation to name our classes.
ex: class Car

الموضوع: _____ التاريخ: / /

* OOP has 4 principles applies them to classes that objects will be instantiated from.

1- Encapsulation
2- Inheritance
3- polymorphism
4- Abstraction

1- Encapsulation = protection
we have to secure our class data.
so encapsulation must be applied and to achieve it we will make 3 steps >>

[1] make variables = Fields as private using underscores
ex: int _age = 20;

[2] make a setter to be able put a value in this field.

[3] make a getter to be able to use this field.

التاريخ: / / الموضوع:

2- Inheritance = no redundancy
= say yes to
Reusability

when having two entities have somthing's in common (fields and methods) and other somethings in differ.
we make a parent class holds all the common members and those entities becomes children of it.
In Dart we use extends class to make a subclass.

---

* Inheritance in Dart is single inheritance. but, we can achieve the multiple inheritance like other programming languages by using the mixin and with keywords.
* one class can apply any number of mixins.

التاريخ: / /

الموضوع: _____

* when we have class B that extends class A. then class C comes to extends class B.

A
↑
B
↑
C

this is like a chain of inheritance called multilevel inheritance

3- polymorphism= many faces
IF we have a superclass called Animal with method named eat( ), two subclasses Fish and Lion inherit this Animal class. Because the two types Fish and Lion eat different food we overide the general eat( ) method in the superclass.
ex: Fish eats Herbs
Lion eats meat

after each sub class has its own implementation, is still can be treated as Animal object.
we can do this ↓ ↓

```
Animal a1 = Fish();
Animal a2 = Lion();
```

so Animal class has two faces.

⊛ method overriding
VS
method overloading

» method overriding achieved by using @override keyword when we need to use the same method in the superclass but with different customaized implementation.

التاريخ: / /

الموضوع: _____

» method overloading
having many versions of the same method only in its name is common, but differs in the return type or even the number of parameters and their type.

4- Abstraction
when I need to force any class that inherits from the super class to implement all methods with its own touch.
the solution is to combine all these methods in an abstract class with no implementation.
And when any class inherits me, it must override all the abstract methods.

* An Abstract class can not be instantiated.

because by Logic It is not useful to make an object that has a method without body. :')

⊛ Interfaces

is a class has methods. although those methods has implementation, but any class implements the Interface must override all methods.

» Interfaces can be instantiated.

⊛ Constructors

From its name, class constructor is responsible for construction of the object.

Constructor is a special method has the same name of the class.

There are default constructor and named one.

الموضوع: _____ التاريخ: / /

- a class may have static members that are shared with all the objects.

Those kinds of members called with the class name.

There is static constructor which called factory, it returns an instance of the class

التاريخ: / /

الموضوع:

Immutable variables

Final

- Can take a value at runtime
- Can take a value at compile time.
- assigns only once
- using if we waiting data From the database and once it comes, it should not changed.

Const

- must take a value at compile time which improve the performance.
- assigns only once

الموضوع: _____ التاريخ: / /

Mixins

"Reusable code blocks"

Imagine this scenario:

IF we have class Animal with methods: sleep() and drink()
→ there are 4 sub classes inherit it. Fish, Shark, Cat, Lion
All those sub classes need the two methods because any animal will sleep and drink.
But there is a problem! :'(

There are only two sub classes Fish and shark need method swim()

And, There are only two other classes Cat and Lion need method walk().

IF we put swim() , walk() in the Animal class this will lead to a problem that the Lion sub class inherits swim() which is logically False.

So, the solution is the mixin concept.
which the only suitable class can inherit what it needs.