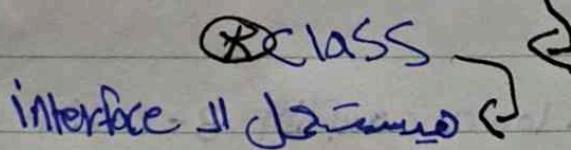


video 04

Interface Segregation

* Split interfaces that are very large into smaller and more specific ones, so that client only have to know about the methods that are of interest to them.



لكل جزء من البرنامج كلاس انتهاي يحتوى على مجموعه من الاجزاء التي تخدم المنهجات المطلوبة من قبل البرنامج.

Ex: Animals class has many methods like eat, sing, walk, run, sleep, fly.
 If we want to implement flying method for birds class, we can't implement all the methods of Animals class. Instead, we can create a new interface called Flyable which contains only the fly method and let birds class implement this interface.

video 05

Dependency Inversion

* High-level modules should not depend on low-level modules, Both should depend on abstract

* Abstraction should not depend on Details, concrete implementations (classes) should depend on abstractions

الأمر

SOLID

التاريخ: ١٢/٣/٢٠٢٣

موضوع الدرس:

Video 01 No astiu

Single responsibility!

- ① Class should only have one responsibility.
it should have only one reason to change

Pros

- 1- testing
- 2- lower coupling
- 3- easier to understand

Open-Closed
video 02

- modules or
④ classes should be open for extension, but closed
for modification

abstract and Polymorphism

Liskov Substitution
video 03

- abstract class
نحو فئة ابجعود وكل
overrides shape كأنه ياتي
وأنا كنت أرجو
Polymorphism كنه ياتي
من الأدوات المطبع والبساطة
are methods فيها interface له
of the desirable properties
of that program.

ويجعله على يد المكتول
الامر