ما هو ال SOLD

هي كلمة مختصرة لخمس مبادئ اساسية بتطبيق هذه المبادئ تجعل الكود قابل للتوسع

والتحديث والاختبار

حرف ال S ==> مبدا المسئولية الواحدة Single Responsibility Principle

حرف ال O ==> مبدا الفتح والاغلاق Open-Close Principle

حرف ال L ==> مبدا الاستبدال Liskov Substitution Principle

حرف ال I ==> مبدا فصل الواجهات Interface segregation Principle

حرف ال D ==> مبدا انعكاس التبعية Dependency Inversion Principle

الفرق بين ال Design Principle & Design Pattern

Design Principle:

هي مجموعة مبادئ وقواعد توضح لنا شكل الكود علشان الكود في النهاية يكون

Extensible, Maintainable , Testable

هذه المبادئ بتجنبنا من العادة السيئة بتاعت التركيز علي الوصول للحل دون الاخد في الاعتبار جودة الحل

ثانيا Design Pattern :

هي حلول عملية وفعلية تم بناؤها علشان تحل مشكله برمجية معينة

الغرض الاساسي منها تقديم حلول برمجية لمشاكل شائعة

Design Principle & Design Pattern الفرق بين ال

ذي الفرق بين الحاجة ال بيقولها المهندس والحاجة ال بيفعلها البناء

المهندس بيعطي التعليمات والتوجهات علشان البناء يكون طبقا لمعايير هندسية معينة Design principle

البناء بينفذ الحلول الفعلية علشان تتم عملية البناء Design Pattern

Coupling , Cohesion , Abstract ,Concrete الاربع مصطلحات دول مهمين

ال Coupling الارتباط

لما يكون في كلاس بيستعمل اوبجكت من كلاس اخر بنقول ان الكلاسين دول مترابطين ممكن الارتباط

يكون ضعيفا : Loose Coupling

يكون قويا : Tight Coupling

كلما كان الارتباط بين الكلاس وبعضها قوي كلما كان الكود صعب الصيانة والتعديل

ولو الارتباط ضعيف بيكون افضل لاننا بنقلل الاعتمادية

ال Cohesion التماسك

هو بيعكس العلاقه بين المكونات

يعني مينفعش الاقي كلاس للموظفين وفي نفس الكلاس بيانات الشركة وبيانات تسجيل الدخول كدا

المكونات غير مترابطة ببعض ....لازم احط الخصائص ال شبة بعض في كلاس واحد

التجريد Abstract & الواقعية Concrete

هوضح المفهومين دول بمثال لما اقول حيوان ده حاجة مجردة انت دلوقت مش هتعرف اي حيوان اقصدة كدا

يعتبر Abstract

لاكن لما اقول قطه اوكلب كدا انت عرفت عن اي حيوان اتكلم ده يعتبر Concrete

لما يكون هنان كلاس في الخصائص الخاصة بتاعته واقدر اخد منه اوبجكت كدا انا بتكلم عن حاجة

واقعية Concrete

لما اتكلم عن حاجة مجردة لا يحتوي علي اي تفاصيل ذي ال Abstract & Interface كدا بتكلم عن

حاجه مجردة Abstract

١- المبدا الاول Single Responsibility

معناها ان كل Function او Class او Model له مسئولية واحدة فقط يقوم بيها وله سبب واحد فقط فى التغيير

مثلا المحاسب هو ال بيقوم بعملية الحسابات في الشركة (مسئولية واحدة)غير مسئول مثلا عن التسويق

كمبدأ المسئولية الواحدة بيجعل الكلاس اكثر تماس Cohesive

متي نحتاج الي هذا المبدا (مبدا المسئولية الواحدة)؟

لما الاقي كلاس معين يقوم باكثر من مسئولية في نفس الوقت.

ذي مثلا كلاس واحد يقوم بقراءة البيانات وحفظ البيانات في قاعدة البيانات والتحقق من سلامة البيانات

ده عبارة عن انذار لاستخدام هذا المبدا واقسم الكلاس ده الي كلاسات صغيرة كل كلاس مسئول عن

مسئولية واحدة فقط

الحل

هعمل كلاس يكون مسئول عن قراءة البيانات ... وكلاس مسئول عن حفظ البيانات في قاعده البيانات

وكلاس مسئول عن التحقق من سلامة البيانات وكدا انا طبقت مبدا Single Responsibility

٢- المبدا الثاني Open-Close Principle

معناه ان بيسمح بالاضافة وغير مسموح بالتعديل

مسموح بالتوسع والاضافه Open > Extended

غير مسموح بالتعديل علي الكود الحالي Close > Modified

ممكن استخدامه هذا المبدا باستخدام ال Abstract & Interface لانه بيسمحوا اننا نضيف علي الكود اللى موجود بداخلة

٣- المبدا الثالث Liskov Substitution Principle

لو عند كلاس للاب وكلاس للابن اقدر اتبادل الادوار بنهم من غير ما ابوظ البرنامج

لو عندي اتنين كلاس S و كلاس لإ ممكن اخلى الاوبجكت بتاع ال S يساوي الاوبجكت بتاع ال T

Ex:

T t = new T( ); S s = new S؛( )

ممكن اقول

T t = new T( ); Or T t = new S؛( )

معناها لو الاب مش موجود ممكن الابن يقوم بنفس الدور

٤- المبدا الرابع Interface Segregation Principle

مخليش مثلا كلاس اضع فيه دوال Non Implemented هو مش المفروض يستخدمها

مثال: انا لما يكون عندي Interface طبيعي لما اي كلاس مثلا يورث منه لازم يطبق كل الدوال ال داخله

طيب ممكن الكلاس مش عاوز يطبق كل الدوال ال فيه في الحالة دي بنستخدم مبدا ISP

وهي بنقوم بوضع الدوال ال محتاجها الكلاس في Interface لوحده

٥- المبدا الخامس Dependency Inversion Principle

معناها ان ال High Level Module مينفعش يعتمد علي ال Low Level Module لازم يعتمدوا الاتنين على ال Abstract

مينفعش ال Abstraction يعتمد على تفاصيل ال Concrete و العكس صحيح

ما الفرق بين ال Low Level Module و ال High Level Module

High Level Module >> هو ال Module ال بيعتمد على Module اخر

Low Level Module >> هو ال Module ال مش بيعتمد على Module اخر او بيتعمد عليه لكن فى اضيق الحدود