12/14/2022

Abdo Gamal Ahmed

no campany

DP

Part3



Table of Contents

[1 –solid principle: 2](#_Toc122373379)

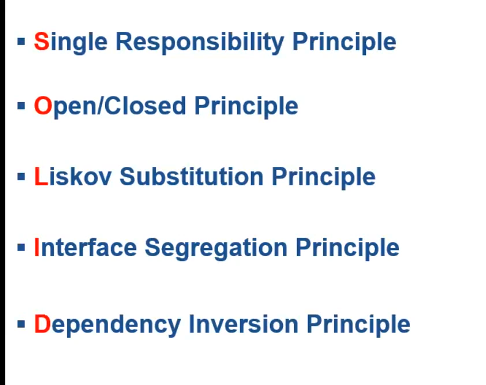
[2 –single responsibility principle : 2](#_Toc122373380)

[2.1 – loose coupling vs. Tight coupling: 3](#_Toc122373381)

[2.2 –separation of concerns : 3](#_Toc122373382)

[2.3 – cohesion : 3](#_Toc122373383)

# 1 –solid principle:

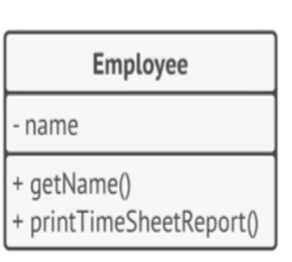


النهاردة هنتكلم عن كل واحد بالتفصيل من دول :

# 2 –single responsibility principle :

من اسمه كدة معناه كل واحد يخليه ف ورقته , كل واحد يعمل حاجة واحدة بس . ودة على مستوى ال class و ال function .

وعمنا بوب بيقول : "a class should have just one reason to change " يعنى الclass الواحد لازم يكون عندة سبب واحد بس عشان يتغير السبب دة يبقا مرتبط ب ال business logic بتاع ال class . بمعنى لو عندى class employee وبعد شويه قلولك عوزين نزود property اسمها addressمكنتش موجوة جوة ال class قبل كدة . ف دى حاجة تخلينا نفتح الكلاس ونغير فيه .

* ف لو انت خليت الclass من الاول responsibility لمجموعه حاجة متخصهوش اصلا ف كدة انت معرض انك تفتح الكلاس اكتر من مرة عشان تعدل فيه لو فيه حاجة جديدة اضافت .
* ****ف كل مكان الكلاس مسئول عن حجات اقل مجرد الحجات الى تخصه ف كدة نسبه التعديل فيه هتقل .

**الهدف من ال principle دة هو reduce complexity**

**تعاله ناخد مثال :**

ف الclass دة هل مطبق مبداء ال single responsibility ؟ لا

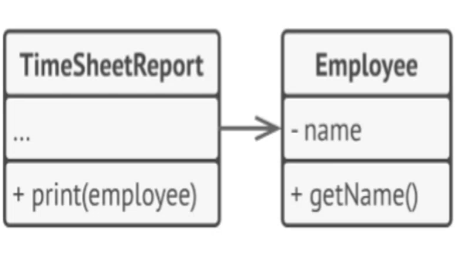
والسبب لو مشيت على كل الحجات الى جوة هتلاقى

* ال name ده هتلاقيه دة من الحجات الاساسيه بتاعه ال emp
* ال get name هتلاقيه دة من الحجات الاساسيه بتاعه ال emp
* اما ال print دى ملهاش علاقه ب ال . emp ممكن تقلى انك بتطبع الداتا بتاعه ال employee

هقلك تمم بس ف فرق بين انك تطبع الداتا بتاعه ال emp وان ال action يكون

بيعمله ال emp يعنى احنا هنا بنتكلم هل الطباعه دى من action الى بيعمها ال emp class ؟

اكيد لا

* ف لو ال print دى موجوة جوة ال employee كل متغير حاجة ف نوع الطباعه هتفتح ال emp وتغير فيه .وحتى لازم الجمله الى قلتها "**كل مغير حاجة ف الطباعه “** ف التغير ملهوش علاقه ب ال emp .
* ف الخلاصه ان ال emp مش مسئول عن ال action ده ولازم الاaction يطلع برة ال employee .

نرجع ناكد على شويه مفاهيم من خلال ال single responsibility بقدر احققهم وهم :

* loose coupling vs. Tight coupling
* separation of concerns
* cohesion

## 2.1 – loose coupling vs. Tight coupling:

احنا هدفنا دايما نعمل تصميم فيه loose coupling يعنى ارطبات الdata او ال class مش مرتبطه ببعض او مش معتمدة على بعض .ودة حققناه لما عملنا interface layer . وزى مقلنا كل مكانت الclass معتمدة على بعض بشكل كبير دة هيوصلنا لصعوبه التغير لو حبيت اغير عشان لما اغير ف class معين هغير برضو ف باقى الclass الى معتمدة عليه .

## 2.2 –separation of concerns :

ودة معناه كل واحد يبص ف وراقته وكل واحد يعمل حاجته .يعنى ف المثال الى فات هفصل ال responsibility بتاعه ال emp عن ال responsibility بتاع الطباعه .

فدايما هنعمل separation of concerns عشان كل كلاس يكون مسئول عن الحاجة بتاعته بس .

## 2.3 – cohesion :

يعنى كميه الترابط بين ال member الى موجوة جوه الكلاس. بمعنى لازم ال member الى جوة الكلاس يكون فيها بينهم علاقه او صله بشكل او باخر .يعنى هتلاقى ال function الى جوة الكلاس بتشتغل على ال member variable الى جوة ال class .

لو لقيت مثلا عندى كام function بيشتغله على 2 او 3 من ال attribute الى عندى وسيبين الباقى .و2 او 3 من ال function التانين بيشتغله على 2 او 3 من ال attribute الى عندى وسيبين الباقى . والكلام دة بيتكرر معاك ف كل ال function الموجودة جوة ال class . ف هنا لازم ال function دول يتفصله ف كلاس لوحهم .