47. Обектен дизайн : принципи на SOLID, open/closed принцип, принцип на регламентираната отговорност.

Solid е популярен акроним, съдържащ инициалите на 5 принципа на софтуерния дизайн: Single responsibility(Регламентирана Отговорност), Open/Closed , Interface segregation , Dependency inversion и Liskov substitution principle.

Open/closed принцип:

обектите(методи/функции/класове) трябва да са отворени за разширяване, но затворени за модифициране.

Single Responsibility Principle-регламентирана отговорност.

В тази легламентирана отговорност трябва да стане ясно, че написаните класове и

методи трябва да им обърнем гръб , и да сме сигурни, че те изпълняват работата, за

която са предназначение и че няма да ги променяме.

Класът трябва да има само една единстветна причина за да го променяме.

Единствения начин да се напише най-лесно клас, който после няма да бъде променян е

класът да има единствено предназначение(изчисление, заявка и т.н.)

Смисълът на принципа не е само да се пишат по-малки класове и методи. Смисълът е в това, че всеки клас трябва да имплементира съвкупност от свързани функции. Един лесен начин да следваме принципа е постоянно да се питаме дали всеки метод и операция на някой метод, е директно свързан с името на класа. Ако намерим методи, които не отговарят на името на класа, трябва да преместим тези методи в друг клас.