30.09.2023 Fatih Hoca

**SOLID**

* **Single of Responsibility == Bir nesnenin tekil bir sorumluluğu olmalıdır.**
* **Open Closed == Proje geliştirilebilir, gelişebilir ama değişemezler.**
* **Liskov == Birbirinden türeyen nesnelerin ataları kapsayıcı olmalıdır ve dönüşüme izin vermelidir.**
* **Interface Segregation**
* **Dependency Injection == Bağımlılıkları tersine çevirebilirsin. Bugün A nın içerisinde B kullanırken başka bir zaman B nin içinde A kullanabilirsin.**

Bakım maliyetlerini minimize etmek ve sürdürülebilir kod sağlamak.

Bir projeye başlamadan önce gereksinimler bulunmalıdır.

MS SQL Database, Business Logic,

Entities, Data, Presentation, Business

Yapısal taraf =Infrastructure Databse e kaydetmek, mail etmek

Entities ==Verileri manipüle ettiğim ve bunları barındırdığım alan.

Application == Business Katmanı Projenin nasıl çalışması gerektiğini yani kural setlerini belirlediğim yer.

ORM RDBS Projedeki class ları database arasındaki tablolar arasında bir ilişki kurmaktır. Yani extend etmektir. Extension

Repository Pattern == Üst katmandan entitylerle yaptığımız eylemlerin sonuçlarını database e taşıyor. Database veri yazan veri gönderen pattern

Repository katmanında contracts buluınmalıdır. ++

***11.10 IOC Dependency Injection***

Operational nesneler yani Operasyon yürüttüğümüz sınıfları biz nerde kullanıcaksak ilk önce bunları havuza veriyoruz. Sonra havuzdan alıyoruz.

Inversion of control bir designdir. Yaratışsal nesneleri belli bir alanda tutun o alandan yönetin der.Merkezi alanda tutun ve oradan dağıtın.

Service localator ve factory patternin birleşimidir.

Ioc ile Uygulama içerisindeki obje instance’larının yönetimi sağlanarak, bağımlılıklarını en aza indirgemek amaçlanmaktadır. Dependency Injection Inversion of Control’ün bir uygulama metotudur.

Dependency injection ile yani bağımlı olduğum nesneleri içeri alma

Constructor,

Method,

Property ‘den dependency injection almak

Bunlar dependency injection yani bağımlı olduğum nesneleri içeri alma

3 tane saklama propertyisi var

* Singleton ==Tekil, Tüm yaratılış süresi boyunca sadece 1 kez yaratılır ve isteyen herkese onu verir.
* Scopped == Bir request boyunca sadece 1 kez yaratılır. O requestin akışı boyunca nerde istenirse o servis bize o servis verir.Tek request bazında
* Transient == En maliyetlisi Projenin her yerinde ister tek requestte ister daha fazla her çağırmada tekrar yaratılır.

Yaratılacak nesnelerimi dependency injection ile alıyorum. Inversiton of control IOC patterni entegre etmiş framework o noktada bağımlılıklarımızı enjekte ediyor.

Inversion of control == nesneleri yaratmak için

Inversion of control kütüphaneleri 3 şekilde saklar.