



GEBZE TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

CSE443 - Object Oriented Analysis and Design
HW1 Raporu

Ad Soyad	Lütfullah TÜRKER
Numara	141044050

Part 1

Öncelikle uygulama için tasarlanacak arayüzü ne ile yapacağıma karar vermek için gerekli araştırmaları yaptım ve JavaFX'in daha kapsamlı ve kullanımı kolay olduğunu farkederek onu kullanmaya karar verdim. Uygulama için JavaFX ile arayüz tasarladıktan sonra kod kısmına geçmeden uygulamanın tasarımını oluşturdum.

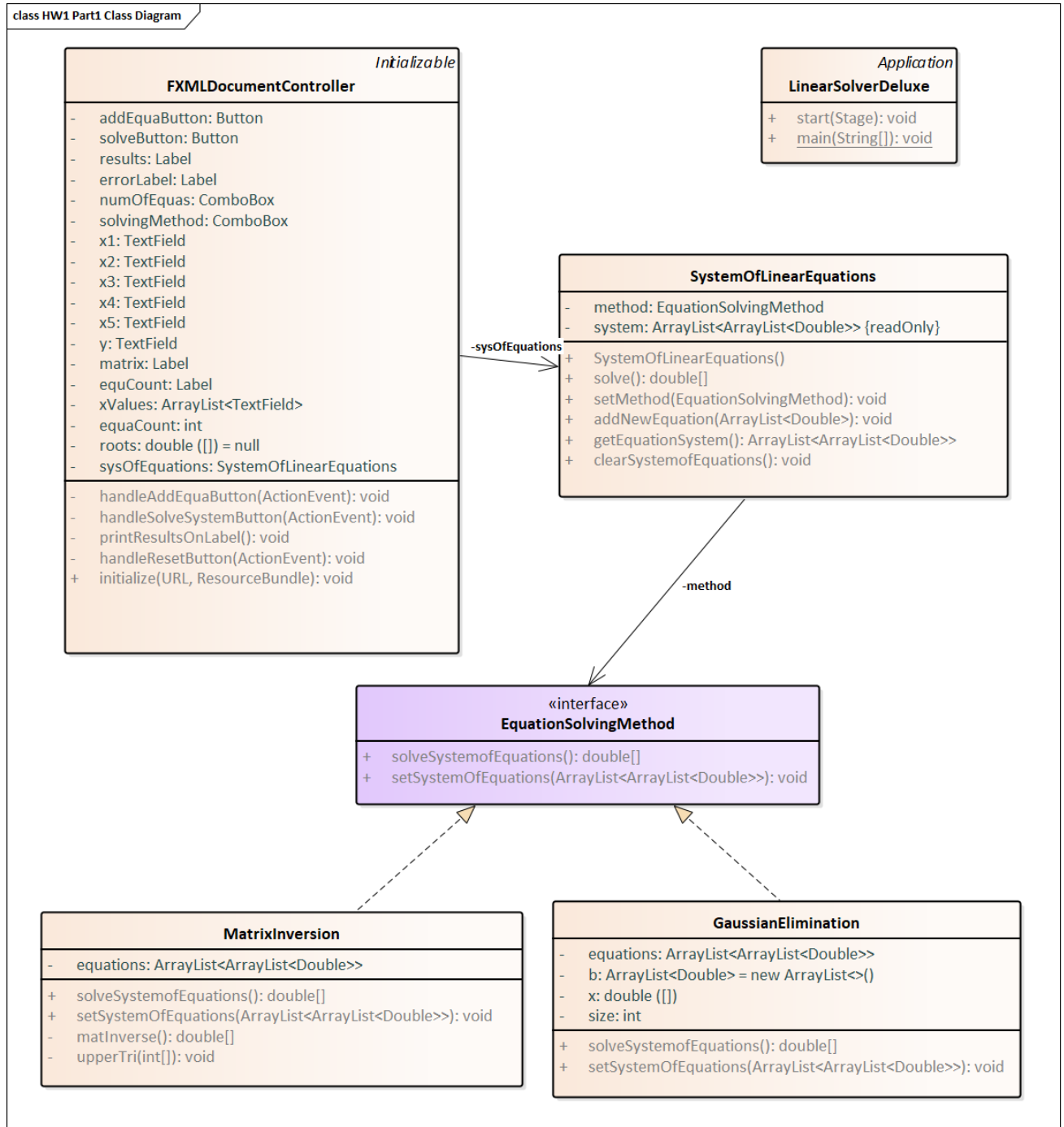
Strategy Design Pattern i kullanarak kullanıcı, denklem sisteminin hangi metod ile çözüleceğine dinamik olarak karar verebiliyor. Bunun için EquationSolvingMethod isimli interface oluşturdum ve tüm çözüm yöntemlerini bu interface i implement ederek yazdım.

Denklem sistemini tutan SystemOfLinearEquations class ında da EquationSolvingMethod tipinde bir variable tanımladım ve arayüzdeki metod seçimine göre bu değişkeni gauss elimination, matrix inversion gibi EquationSolvingMethod alt classlarından hangisi seçildiyse o class'a atayarak çözümün o yöntemle yapılmasını sağladım.

Arayüz için Reset butonu koyarak birden fazla denklem sistemini uygulamadan çıkmadan arka arkaya çözebilmeyi sağladım.

Denklem sistemindeki denklem sayısının seçimine göre dinamik olarak bilinmeyenlerin katsayılarını alan TextField leri açıp kapadım.

Ayrıntılı tasarım için uygulamanın Class Diagram'ı aşağıdadır.



Part 2

Planlanan uygulama için bir abonelik ve abonelere haber verme sistemi olacağından dolayı Observer Design Pattern'ini kullanmayı tercih ettim.

Bir abone birden fazla içerik türüne abone olabileceği için Subscriber Observer sınıfının içinde tek bir abonelik tutmak yerine aboneliklerinin bir listesini tutmayı düşündüm. Abone olunabilir içerik türleri için Enum tuttum.

Yazdığım genel Observable interface'ini implement ederek İçerikler için bir Abstract sınıf düşündüm ve burda tüm içeriklede ortak olacak temel özelliklere yer verdim. (Content Abstract Class)

Tüm içerik türleri bu Content sınıfını extend ederek oluşturulacak.

Daha sonra yeni bir içerik türü eklendiğinde de yine Content sınıfı extend edilerek oluşturulacak ve enum içerisine yeni içerik türü için bir enum eklenecek.

Böylelikle yeni içerik türü eklenmesi için oluşacak bakım maliyeti yalnızca enum içerisine bu içerik türünü eklemek ve içerik türü için Content sınıfını extend ederek bir Class yazmak olacak.

Subscriberler için abone olma,abonelikten çıkma gibi işlemler için de Content sınıfında getContentType() fonksiyonu ile Content tipini içeren enum return edilecek.

Subscriber tarafında ise abone olunan içerikler ArrayListte tutulacağı için arraylist'in hangi elemanın hangi içerik sınıfına ait olduğunu anlamak için getContentType fonksiyonu kullanılacak.

Örneğin 1 den fazla içerik sınıfına abone olmuş bir kullanıcı Audio aboneliğini iptal etmek istiyor ise unsubscribe fonksiyonuna yalnızca enum içerisindeki Audio parametresini vermesi yeterlidir.

Böylelikle ArrayList içerisinde getContentType fonksiyonu Audio return eden elemandan removeObserver fonksiyonu çağrılır ve abonelik iptal edilebilir.

Planlanan uygulamanın Class Diagram'ı aşağıdadır.

