

CSE443 - Object Oriented Analysis and Design

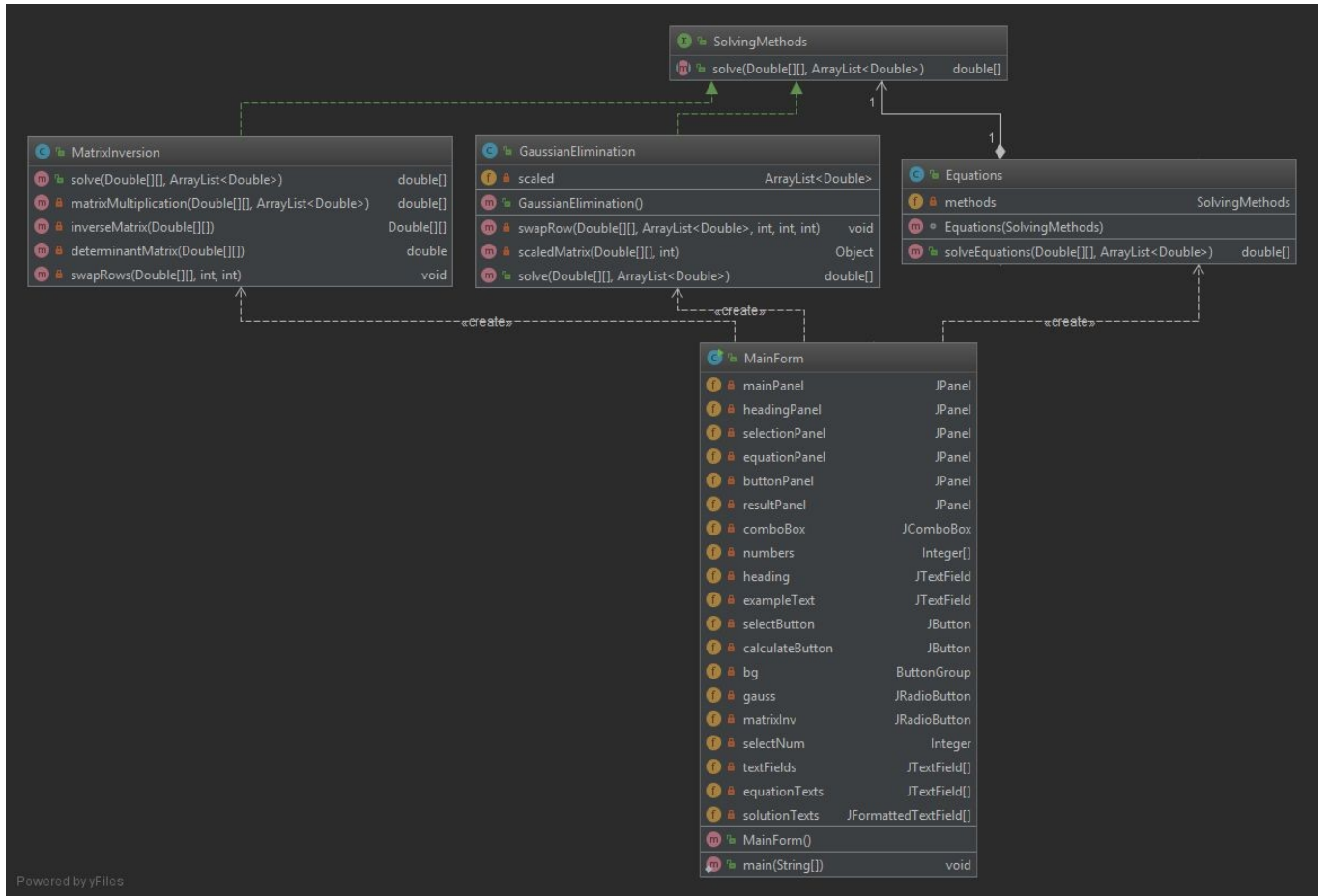
HW01 Report

Emire Korkmaz 141044043

Question 1:

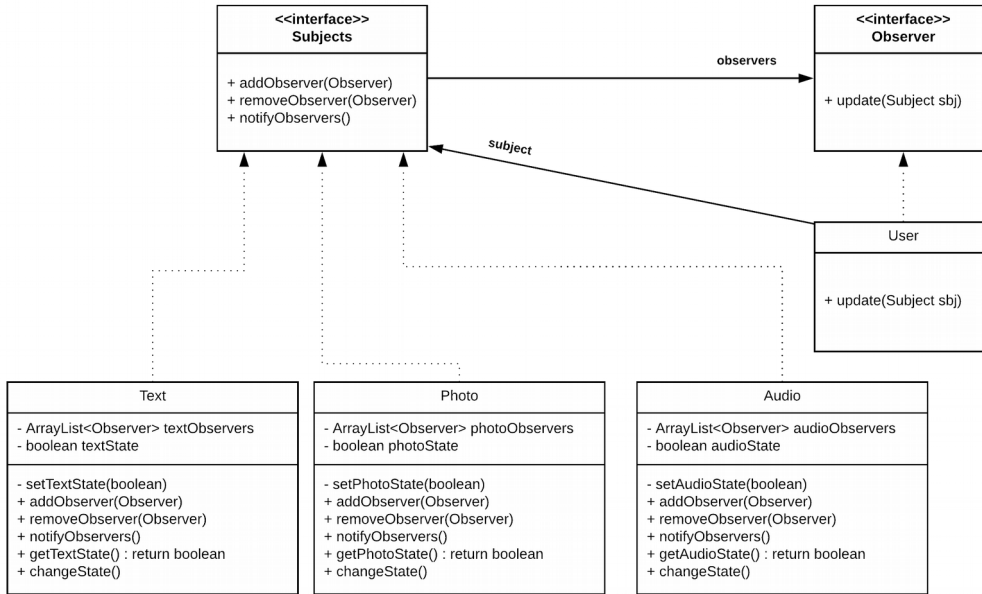
Strateji tasarım desenini seçildi. Çünkü buradaki problemde, bir işlem için birden fazla farklı yöntemlerin uygulanabileceği bir durum mevcuttur ve kullanıcı çalışma esnasında diğer yöntemlere rahatlıkla geçebilir. Esneklik vardır.

Problemde kullanıcıdan gelen lineer sistemin çözülmesi istenmiştir. Bunun için bir “SolvingMethods” adında bir arayüz üretilmiştir ve bu arayüzü implement eden iki tane metod üretilmiştir. (Gaussian Elimination ve Matrix Inversion) Daha sonra bu metodları kullanabilecek “Equations” adında bir context sınıfı oluşturulmuştur. Eğer programa yeni bir method eklenmek istenirse, yapılması gereken tek şey “SolvingMethods” adındaki arayüzü implement eden bir sınıf oluşturulmasıdır.



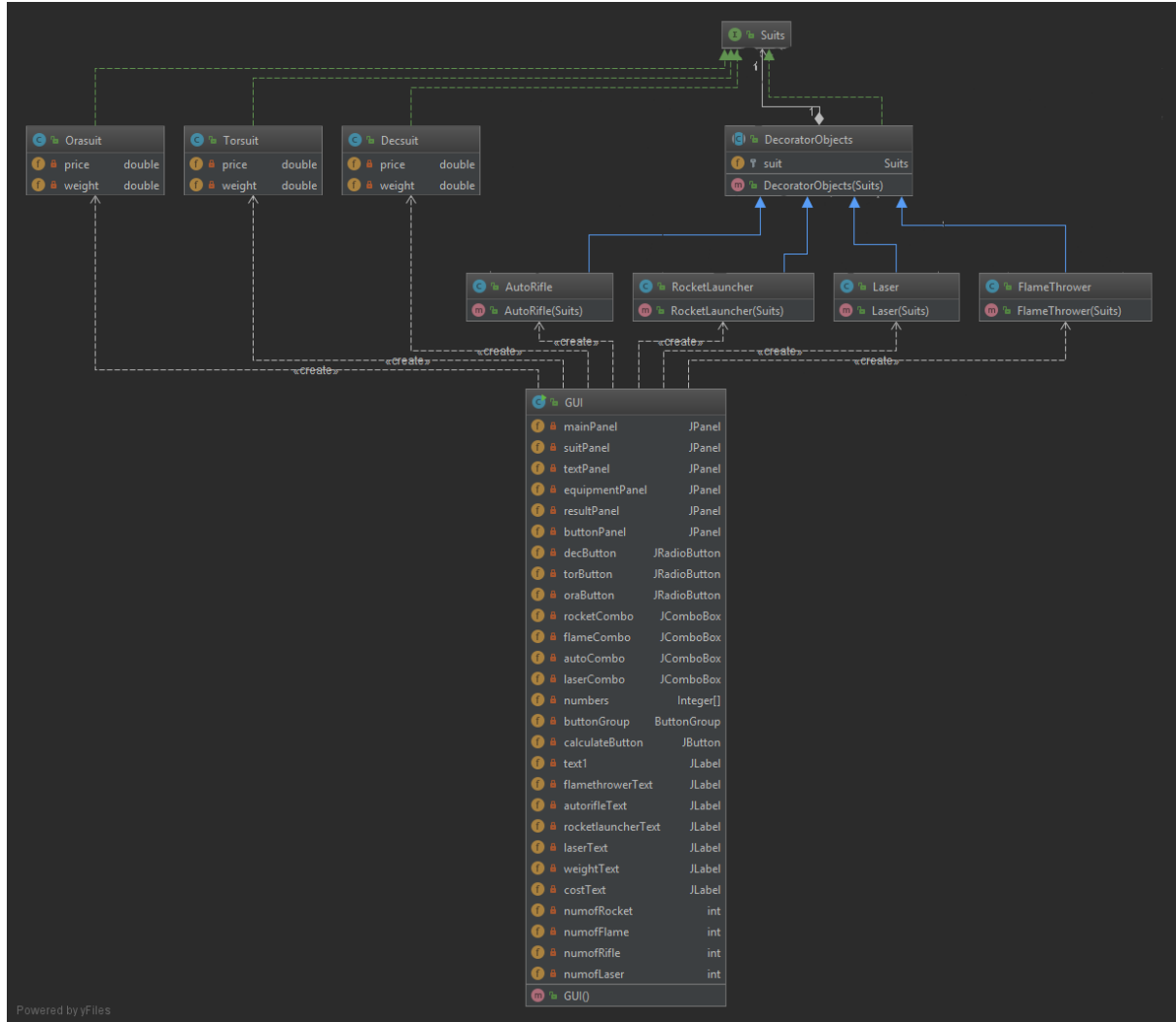
Question 2:

Observer (Gözlemci) tasarım desenini seçtim. Çünkü buradaki problemde, kullanıcı bir web sitesine abone oluyor ve bu web sitenin içerikleri değiştiği anda bilgilendirilmek istiyor. Observer tasarım deseninde one-to-many ilişkisi vardır. Problemin class diagramı oluşturulurken, ilk olarak “Observer” isminde bir arayüz oluşturulmuştur ve bu arayüzü implement eden “User” adında bir sınıf oluşturulmuştur. Daha sonra “Subjects” adında bir arayüz oluşturulmuştur. Bu arayüz takip edilecek nesneleri belirtir. Bu arayüzü implement eden üç tane sınıf (takip edilecek nesneler) oluşturulmuştur.(Photo, Text ve Audio) Programa dördüncü bir tip eklenmek istendiğinde yapılacak tek şey “Subjects” arayüzünü implement etmektir. Bu sayede esneklik sağlanmıştır.



Question 3:

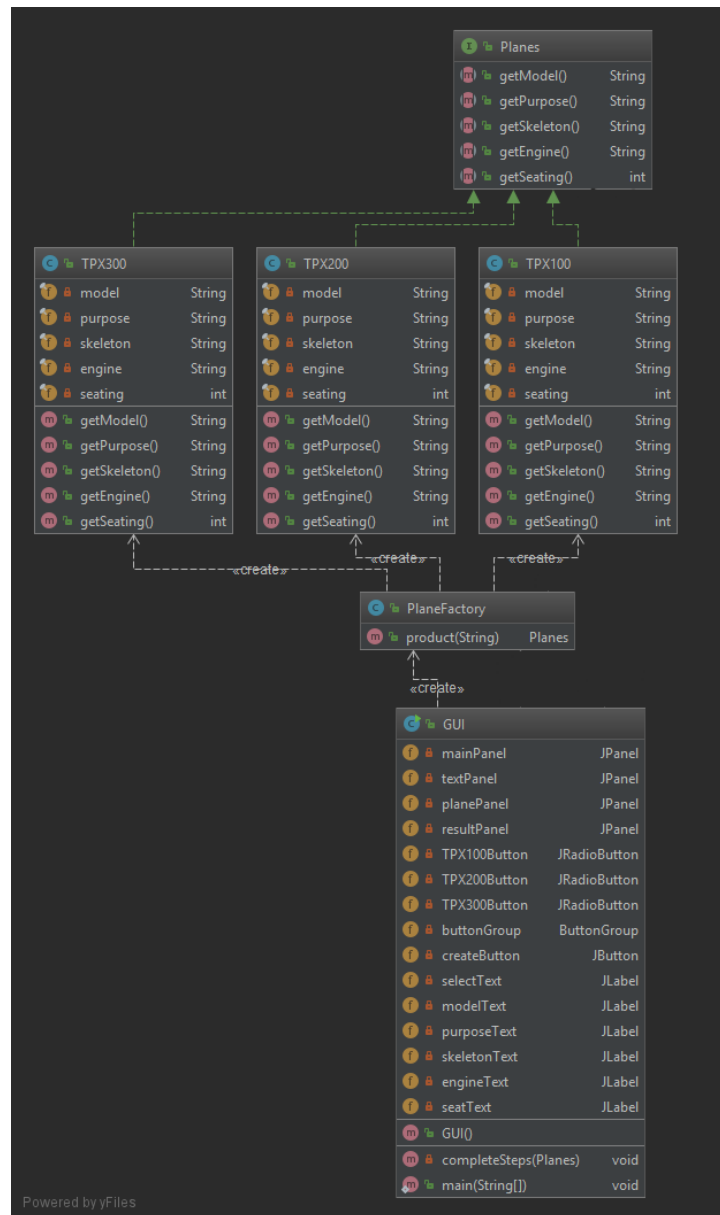
Decorator tasarım deseni Structural grubuna aittir ve bir yapıya dinamik olarak yeni metotlar eklenmesini düzenleyen bir tasarım desendir. Decorator tasarım deseni yapıya yeni metot eklenmesini yapının arayüzünü uygulayan bir decorator nesnesi tanımlanarak yapar. Problemden bizden bir giysiye aksesuarlar takılması isteniyordu. (Flamethrower, autorifle, laser, rocket launcher gibi) Bu sayı birden fazla olabilirdi. Örnek olarak; bir giysiye 2 flamethrower, 1 rocket launcher, 4 laser, 2 autorifle gibi. Bu durumu yapabilmek için decorator tasarım deseni seçildi.



Question 4.1:

Factory tasarım deseni creational grubuna aittir. Amacı ise nesnelerin oluşturulmasına yönelik bir tasarım önerisi sunmaktır. Factory tasarım deseni, istenen tipte nesne oluşturma sürecini Client'ın bu konuda detay bilgi sahibi olmadan gerçekleştirilmesini sağlar. Yeni oluşturulan nesneye bir interface ile referans edilerek ulaşılmasını sağlar.

Problemde farklı modellerde uçak üretilmesi gerekiyordu. İlk olarak, "Planes" adında bir arayüz oluşturuldu. Daha sonra bu arayüzü implement eden üç tane farklı modelde uçak sınıfları oluşturuldu ("TPX100", "TPX200", "TPX300") ve bu uçakları üretecek bir tane fabrika sınıfı oluşturuldu. Kullanıcı farklı modelde bir uçak üretmek isterse yapması gerek tek şey "Planes" arayüzünü implement eden yeni bir sınıf oluşturmaktır.



Question 4.2:

Abstract Factory tasarım deseninde ise farklı ürünler üreten fabrikaların soyutlanması fikri üzerine kurulu bir tasarım kalıbıdır.

Problemde uçakların farklı pazarlara üretilmesi isteniyordu. Bunu yapabilmek için “Factories” adında bir abstract sınıf oluşturuldu. Daha sonra bu sınıfı inherit eden “DomesticFactory”, “EurasiaFactory” ve “OtherFactory” sınıfları oluşturuldu. Kullanıcının, üretilen uçağı başka bir pazara dağıtması gerektiğinde yapması gerek tek şey “Factories” abstract sınıfı inherit eden yeni bir sınıf oluşturmaktır.

