

**Gebze Technical University
Computer Engineering**

CSE 443 - 2019 Autumn

HOMEWORK 1 REPORT

**Akın ÇAM
151044007**

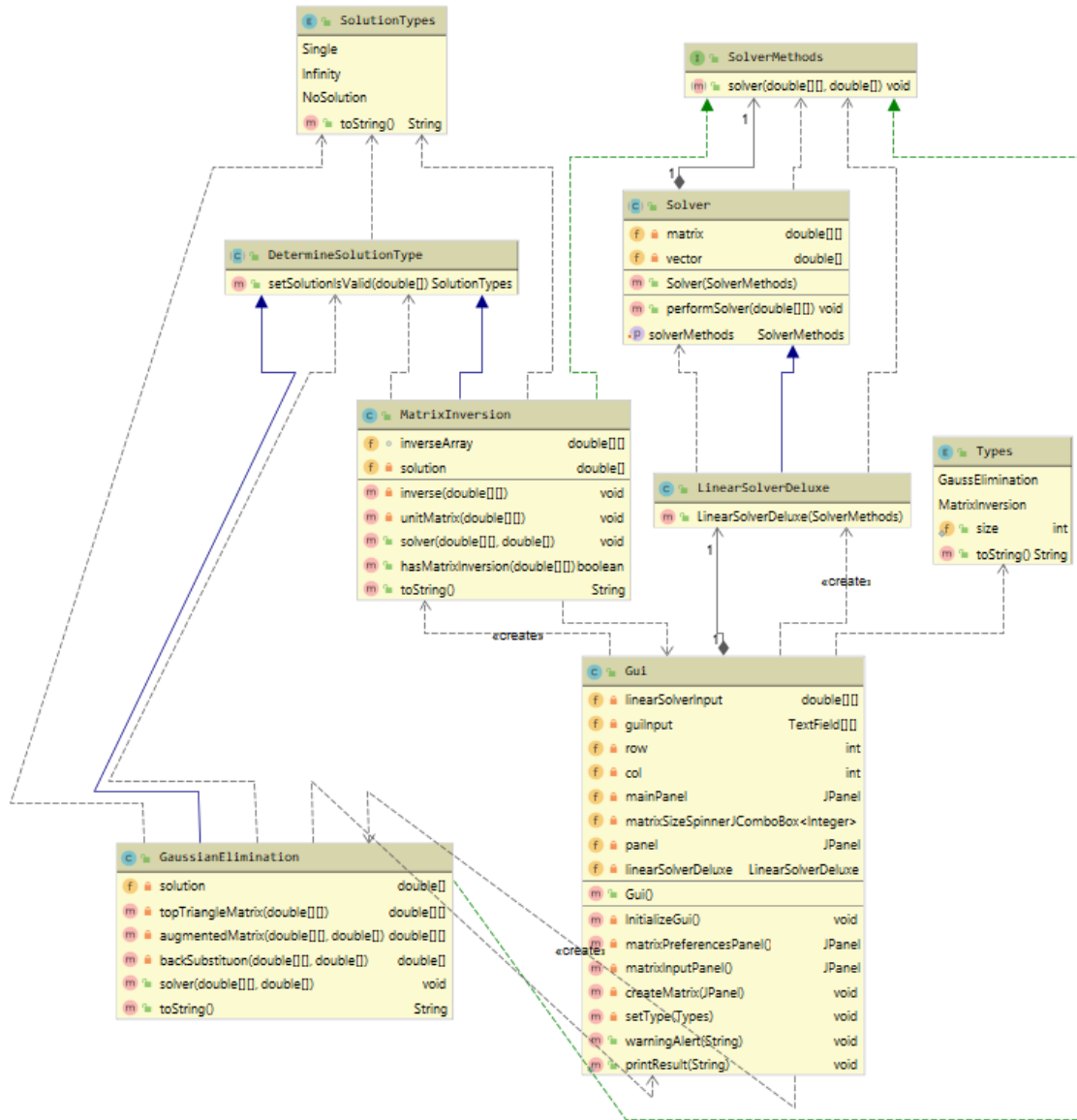
**Course Teacher:
Ercan Aptoula**

IntelliJ İdea ortamında geliştirilmiştir.

Question 1:

LinearSolverDeluxe projesi tasarlanırken Strateji tasarım deseni kullanılmıştır. Strateji tasarım deseni bir işlem için farklı davranışlar olduğunda, bu davranış dinamik olarak değiştirilmek istendiğinde, davranış yöntemi kullanacak nesnenin secmesi istendiğinde ve loosely coupling e uygun olarak yeni davranışları kolayca eklenmek istendiğinde Strategy tasarım deseni kullanılır. Bu bilgiler ışığında linearSolverDeluxe için şimdilik 2 tane çözüm yöntemi bulunmaktadır. Bu yöntem dinamik olarak değiştirilmek istenmektedir. Yeni yöntemlerin kolayca eklenmesi istenmektedir. Tüm bu sebeplerden dolayı Strateji tasarım deseni kullanılmıştır.

Class Diagram :



Burada İki algoritma ailesi bulunmaktadır. Birincisi Client diğeri SolverMethods ve bu sınıfı implement eden sınıflar. Solver methodu abstract olarak gerçekleştirilmiştir. Bu sınıf denklemleri çözme davranışını belirlemek için SolverMethods objesi bulundurmaktadır. Ayrıca bu sınıfta çözüm metodunu dinamik olarak değiştirmek için setSolverMethod(method) metodu

bulunmaktadır. PerformSolver methodu SolverMethods objesinin .solver(...) methodunu kullanır. Gui sınıfı burada kullanılmaktadır. Bu sınıftan LinearSolverDeluxe sınıfı extends edilmiştir.

Diğer algoritma ailesi bir davranış içeren SolverMethods interface sınıfı ve bu sınıfı gerçekleyen sınıflardır. SolverMethods hangi davranışı gerçekleyeceklerini bir method ile belirler ve bu arayüzü gerçekleyen sınıflar bu methodu kendi davranışına göre gerçekler.

GaussElimination sınıfında verilen denklemleri ilk olarak augmented matris oluşturulur daha sonra üst üçgen matris haline getirilir ve backSubstitution ile sonuç hesaplanır.

MatrisInversion sınıfında verilen denklemin alt üçgen matris, üst üçgen matris yöntemleri ile birim matris haline getirilir ve sonuç hesaplanır.

SolutionTypes bir enumdur. MatrisInversion ve GaussElimination hesaplanan sonuçların tipi için (tek sonuç, sonsuz sonuç ve sonuç yok) oluşturulmuştur.

Types bir enumdur. Gui tarafından kullanılır ve kullanıcıya yöntem seçme imkanı sağlar.

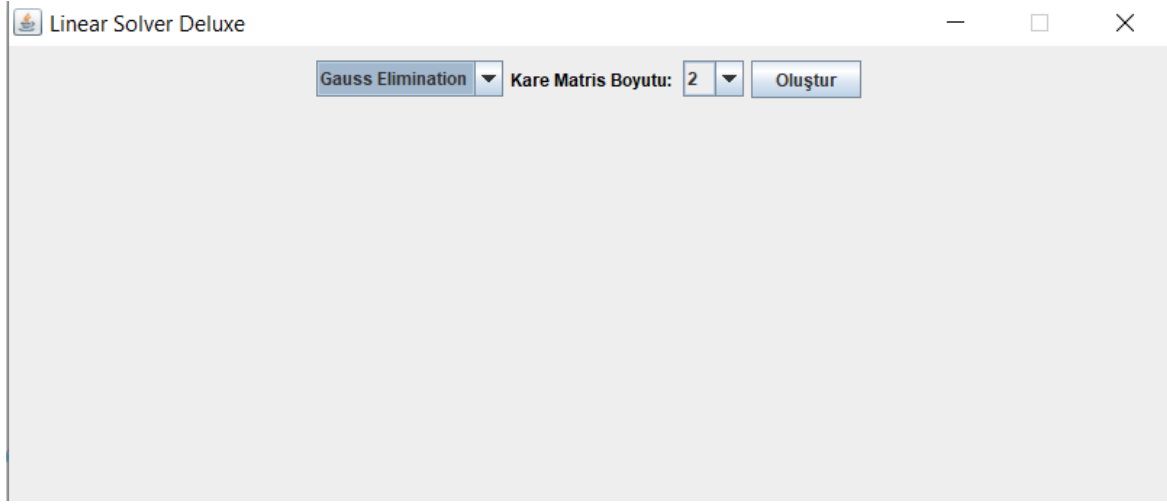
Types bir enumdur. DetermineSolutionType tarafından kullanılır. İşlem sonucunun tipi Types tipinde döndürülür.

Gui java swing ile gerçekleştirilmiştir. 1. panelde Kullanıcıdan bir kare matris alır sayısı alır (lineer denklem kare matris olmalıdır.). 2. panelde denklemleri almaktadır. Daha sonra AlertDialog olarak sonucu paylaşmaktadır.

Yöntem :

Yeni bir yöntem eklemek için yöntem gerçekleştirildikten sonra ilk olarak yöntem adı Types enum sınıfına eklenmesi gerekmektedir. Ardından Gui sınıfında setType methoduna eklenmelidir.

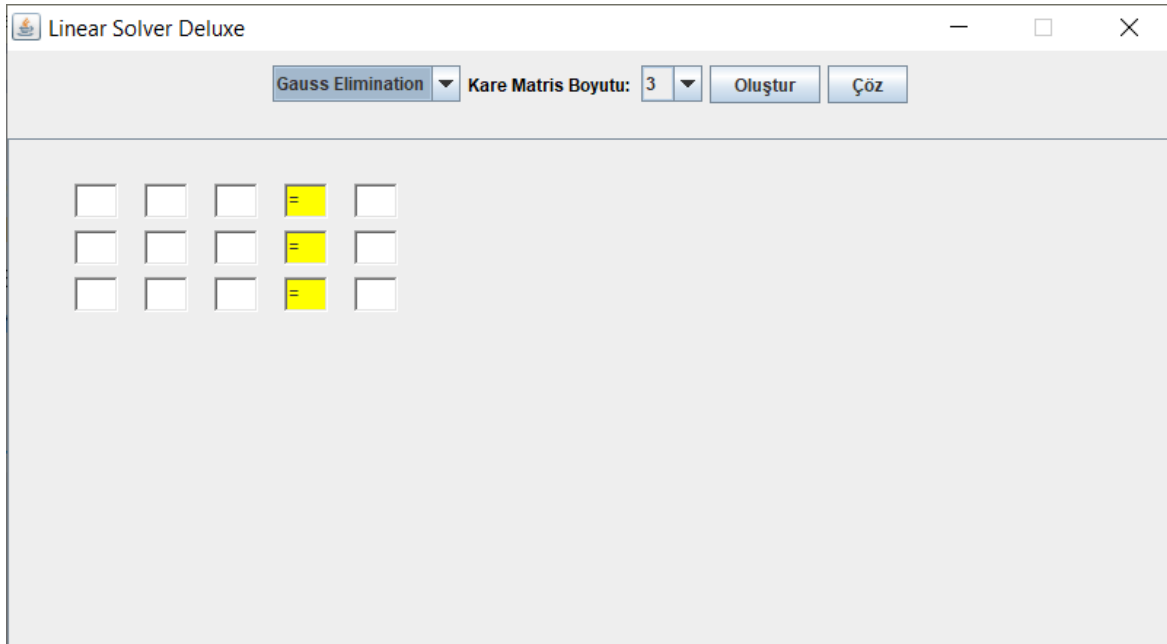
Çıktı :



Linear Solver Deluxe

Gauss Elimination ▼ Kare Matris Boyutu: 2 ▼ Oluştur

(kullanıcıdan matris sayısı alınır)



Linear Solver Deluxe

Gauss Elimination ▼ Kare Matris Boyutu: 3 ▼ Oluştur Çöz

(oluştur butonu ile matris olusturulur.)

Linear Solver Deluxe

Gauss Elimination ▼ Kare Matris Boyutu: 3 ▼ Oluştur Çöz

2	-1	3	=	5
2	2	3	=	7
-2	3	0	=	-3

Solution is-->2.5-->0.666... ×

i Solution is-->
-->2.5
-->0.6666666666666666
-->0.22222222222222224

OK

(işlem sonucu)

Linear Solver Deluxe

Gauss Elimination ▼ Kare Matris Boyutu: 3 ▼ Oluştur Çöz

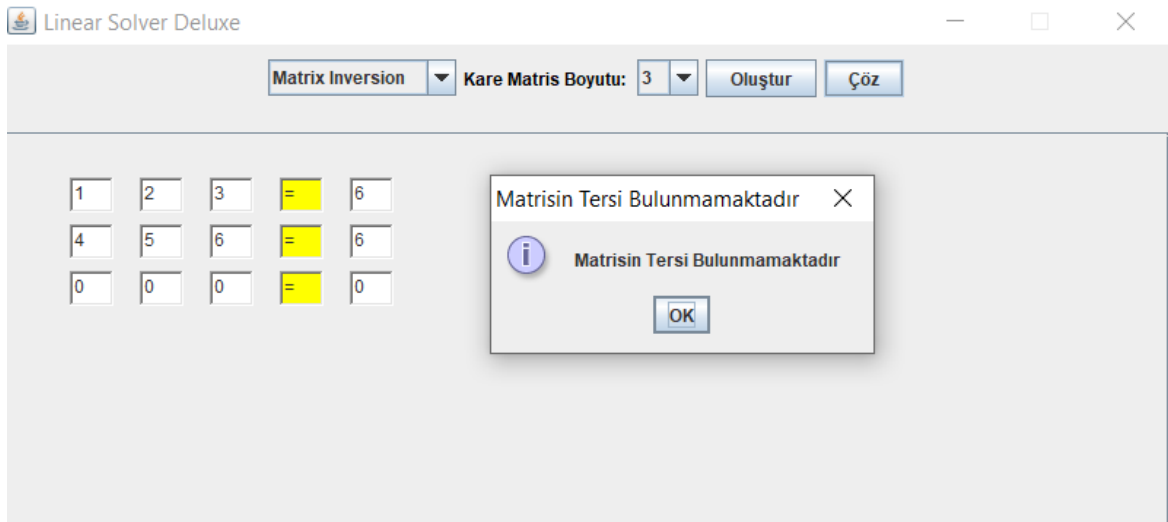
1	2	3	=	6
4	5	6	=	6
5	5	5	=	6

Solution is-->INFINITY ×

i Solution is-->
INFINITY

OK

(işlem sonucu)



(işlem sonucu)

Question 2:

Bu proje için observer tasarım deseni kullanılmıştır. Observer tasarım deseni behavior grubuna aittir ve bir nesnede meydana gelen değişikliklerde içinde bulundurduğu listede bulunan nesnelere haber gönderen tasarım desendir. Burada Contentlerden herhangi biri değiştiğinde ya da bunların kombinasyonları değiştiğinde haber verilecektir. Bir subject birden çok observer olmalı. Bu yüzden Observer tasarım deseni kullanılmaktadır.

2 adet interface bulunmaktadır:

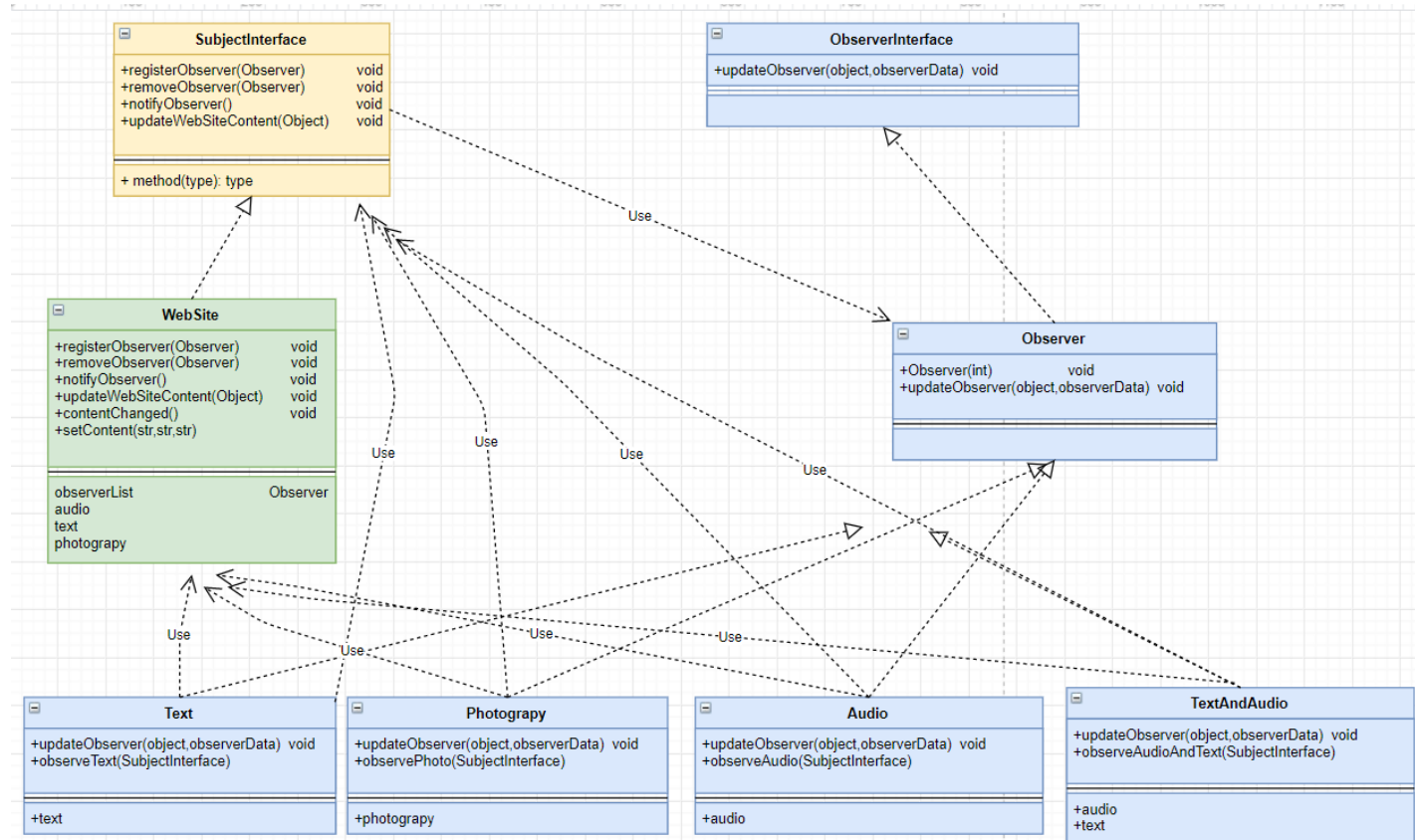
Subject interface ve Observer interface. Subject sınıfı Observer sınıfına etki eden sınıftır. Subject sınıfı değişikliğin meydana geldiği nesneyi içermektedir. Bu değişiklik meydana gelmesi durumunda üyeler uyarılacaktır.

Observer sınıfı bir subject parametresi almaktadır. Bu registerObserver ile kendisini listeye ekler ve update olduğunda uyarılmayı sağlamaktadır.

Contentler ileride arttırılabilir. WebSite subject interface i gerçekleştirir.

Bu Content typeları tutmaktadır. Contentlerde bulunan(Text, Photo etc) observer....(subject) methodu ile kendini ilgili web sitenin ilgili kısmını seçer. Update methodu ile kendini listeye ekler ve değışiklik olduğunda uyarılır

Class Diagram :

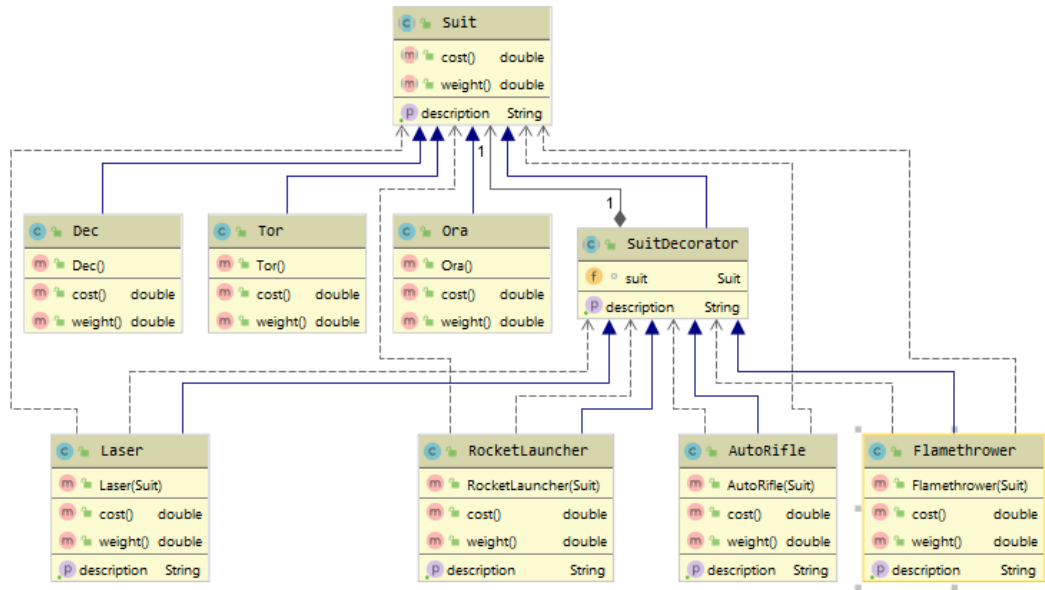


Question 3:

Bu proje için Decorator tasarım deseni kullanılmıştır. Decorator dizayn deseni bir objeye objenin Sınıfına müdahale etmeden yeni özellikler eklemek için kullanılmaktadır. Bu işlemler dinamik olarak eklenebilir. Bu bilgiler ışığında Bir

Suit e yeni bir aksesuar ya da aksesuarlar eklemek için yeni aksesuar tipleri eklemek için bunların ücreti ve ağırlığını hesaplamak için Decorator dizayn deseni kullanılmıştır.

Class Diagram :



Suit sınıfı soyut sınıftır. getDescription(), cost(), weight() metodları soyut olarak bırakılmıştır. Bu sınıftan üç ana Zırh sınıfı gerçekleştirilmiştir(extends). SuitDecorator sınıfı soyut bir sınıftır. Bu sınıf Suit objesini bulundurmaktadır. Bu sınıftan türeyen aksesuarlarda soyut metodlar gerçekleştirilmiştir. Bu işlemler gerçekleştirirken suit objesinden gelen bilgiler ve kendi bilgisi döndürülür. Bu sayede hem ana sınıflarda değişiklik yapılmaz hem de bir sınıfa müdahale etmeden yeni özellikler eklenebilmektedir.

Çıktı :

-----Question 3-----

Dec 500.0k TL 25.0 kg.

Ora, Laser, Laser, Flamethrower 1950.0k TL 43.0 kg.

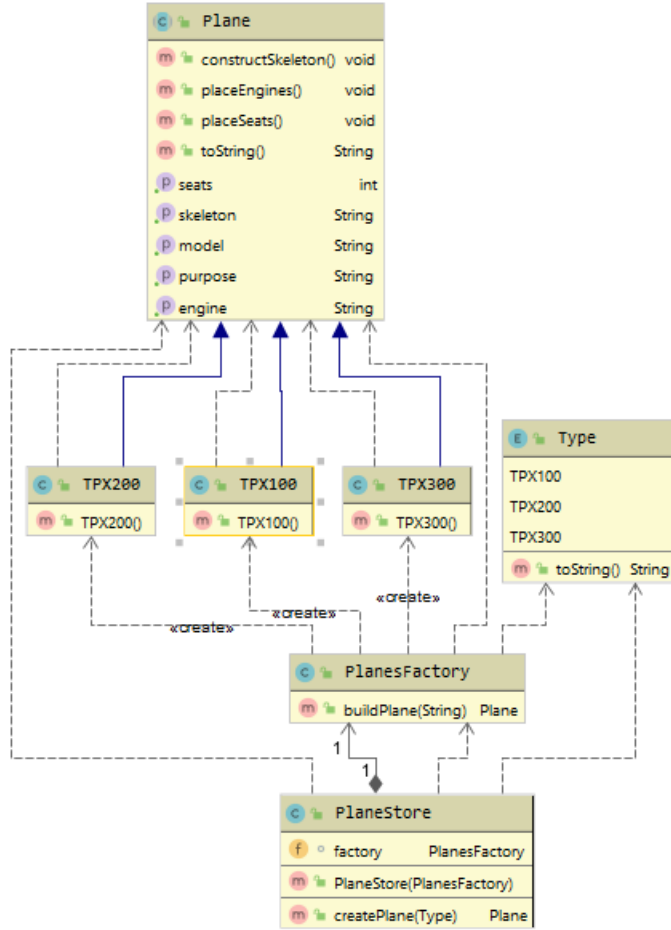
Tor, AutoRifle, Flamethrower, RocketLauncher 35200.0k TL 61.0 kg.

-----Question 4.1-----

Question 4.1:

Bu proje için Factory dizayn deseni kullanarak oluşturulmuştur. Factory tasarım deseni bir nesne yaratımındaki gerçek mantığı gizlemek için kullanılan soyutlama biçimidir. Bu sebeple plane oluşumu gizlenmesi için bu tasarım deseni kullanılmıştır. Uçak modelinin olusturulması bir fabrika sınıfında gerçekleştirilmiş ve gizlenmiştir.

Class Diagram :



PlaneStore Plane oluşturduğumuz sınıftır. Fabrika objesini içerir ve ürünün initialize edilmesini sağlar. Plane, Factory sınıfının ürünüdür. Üç tip gerçekleştirilmiştir(TPX 100,200,300. Lously Coupled olduğu için yeni modeller eklenebilir. PlanesFactory Plane oluşturmayı sağlamaktadır. Type enum sınıfı main de herkes tarafından ortak bir yapıda input girmek için kullanılmıştır.

Çıktı:

-----Question 4.1-----

Aluminum alloy skeleton is constructing
Single jet engine is placing
50 is placing
We create a plane model of TPX100

---- TPX100 ----
Domestic flights
Single jet engine
Aluminum alloy
50

Nickel alloy skeleton is constructing
Twin jet engines is placing
100 is placing
We create a plane model of TPX200

---- TPX200 ----
Domestic and short international flights
Twin jet engines
Nickel alloy
100

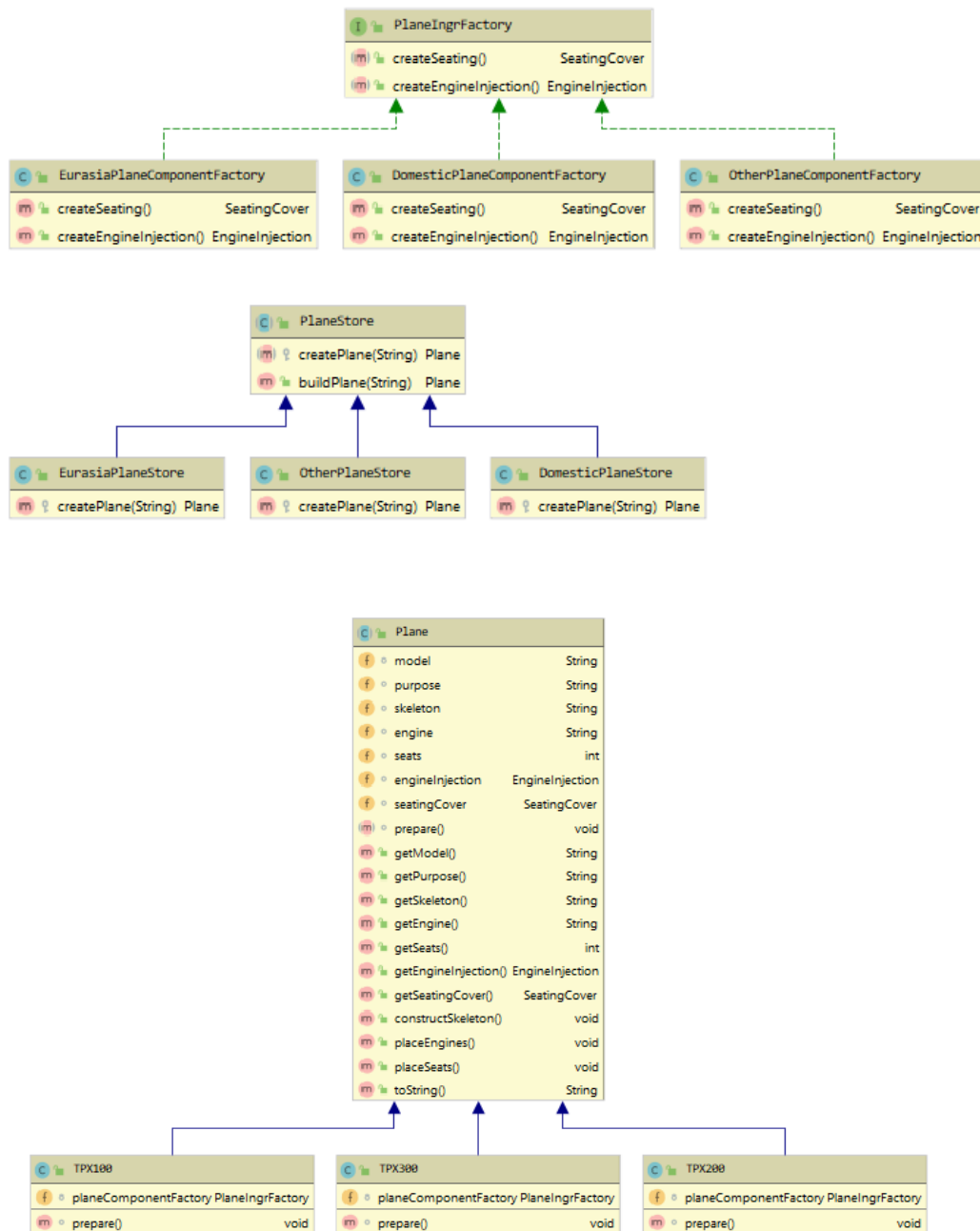
Question 4.2:

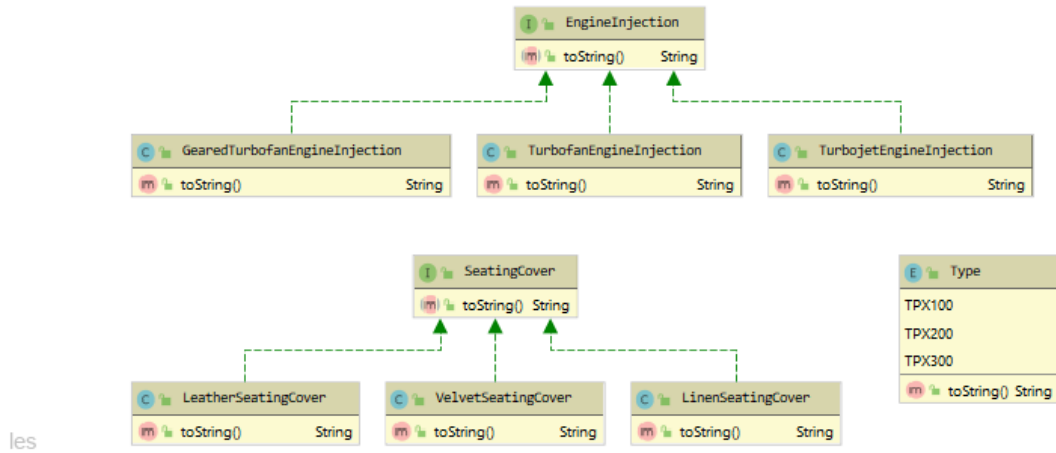
Bu projede Abstract Factory tasarım deseni kullanılmıştır. Abstract Factory tasarım deseni somut sınıflarını belirtmeden ilgili veya bağımlı nesnelerin ailelerini oluşturmak için bir arayüz sağlamaktadır. Farklı ürün aileleri için farklı arayüzler mevcuttur. Abstract Factory farklı farklı ürünlerin üretildiği fabrika olarak tanımlanabilmektedir.

Bu bilgiler ışığında Farklı fabrikalarda farklı tiplerde özellikler eklendiği için farklı marketler bulunduğu için farklı ürün aileleri bulunduğu için Abstract Factory tasarım deseni kullanılmıştır.

Class Diagram :

Not: class diagram çok büyük olduğu için sade hali eklenmiştir. Olması gereken hali Package question4_2 adında proje klasöründedir.





PlaneIngrFactory sınıfı interfacedir ve concrete fabrika sınıfları bu sınıftan gerçekleştirilmiştir. Burada fabrikalarda farklı türlere göre üretilen Koltuk tipi ve motor enjeksiyon metodları gerçekleştirilmek üzere boş bırakılmıştır.

Interface olan SeatingCover ve EngineInjection her bir concrete fabrikası için bir arayüzdür ve bu komponent özelleştirilmiştir.

PlaneStore soyut bir sınıftır ve çalışma zamanında oluşturulmaktadır.

Concrete fabrika aileleri farklı fabrikalardır ve kendilerine özgü tiplerde üretim özellikleri mevcuttur.

Type bir enumdur ve Plane oluşturulurken ortak bir input oluşturulması sağlanmıştır.

Çıktı:

-----Question 4.2-----

|| Building a --> TPX300 ||

Titanium alloy skeleton is constructing

Quadro jet engines is placing

250 is placing

---- TPX300 ----

Transatlantic flights

Quadro jet engines

Titanium alloy

250

Velvet Seating Cover

Turbojet Engine Injection

|| Building a --> TPX100 ||

Aluminum alloy skeleton is constructing

Single jet engine is placing

50 is placing

---- TPX100 ----

Domestic flights

Single jet engine

Aluminum alloy

50

Linen Seating Cover

Turbofan Engine Injection

