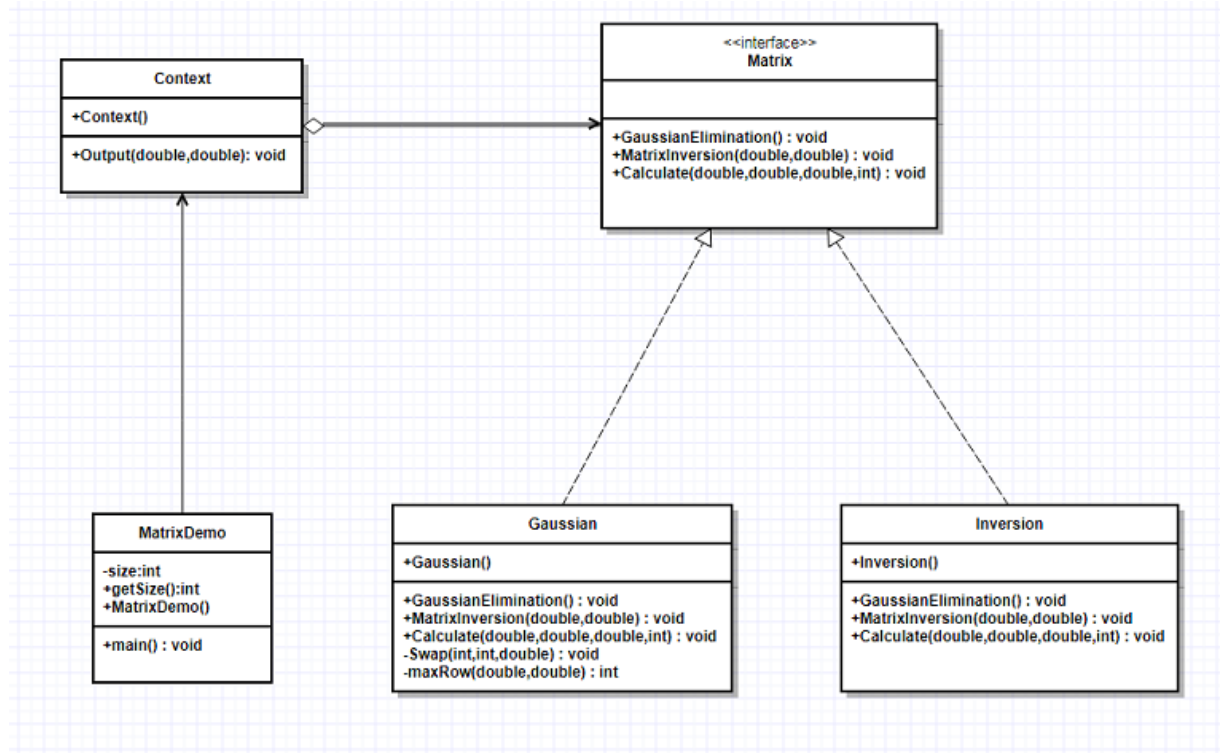


BİL443 - Object Oriented Analysis and Design HW1 Raporu

EDA BAHİRİOĞLU-131044055

Q1 . Ödev programlaması Netbeans Ide kullanılarak yapılmıştır.Uml Q1 için çizilen UML diyagram aşağıya eklenmiştir.Bu problem için Strategy Pattern göz önünde bulundurulmuştur.Aynı zamanda dosya içine de görüntü olarak eklenmiştir.



Diagramda 5 class bulunmaktadır.Matrix İnterface olarak kullanılmış,diğer fonksiyonlar ayrı class larda implement edilmiştir.Diagramın class yapında programlanması zip dosyasının içindedir.

Gaussian.java içinde gaussian methodu için işlemler bulunmaktadır.Ancak tam bir gauss elimination işlemlerine sahip değildir.Gauss işlemleri için kullanılacak rowlar arası swap işlemi için swap fonksiyonu,swap te kullanılacak maximum row fonksiyonu,en son sonucun hesaplandığı calculate fonksiyonu genel Gaussian işlemleri için gaussian elimination fonksiyonu bulunmaktadır.Inversion.java içinde ise Inversion işlemlerinin yapıldığı MatrixInversion fonksiyonu ve sonucun hesaplandığı calculate fonksiyonu bulunur. Context class ile matrix classı arasında aggregation ilişkisi vardır.Bu ilişki output fonksiyonu ile sağlanır.Bu fonksiyon içinde denklemin hangi metodla sağlanacağına dair koşullar vardır.Çözümü olmadığı takdirde hata mesajı verilir.MatrixDemo.java içinde test işlemlerinin yapıldığı main fonksiyonu bulunur.Girdiler için gerekli Exception işlemi yapılmıştır.Program şu adımlar ile çalışmaktadır:

İlk olarak matrix size i istenir.(Bu program sadece 2 boyutlu matrixler için çalışmaktadır.)

Enter matrix size:

2

Enter two coefficient (iki denklem için de iki tane katsayı girilir)

2 5

1 3

Son olarak her iki denklem için sonuç değeri girilir.

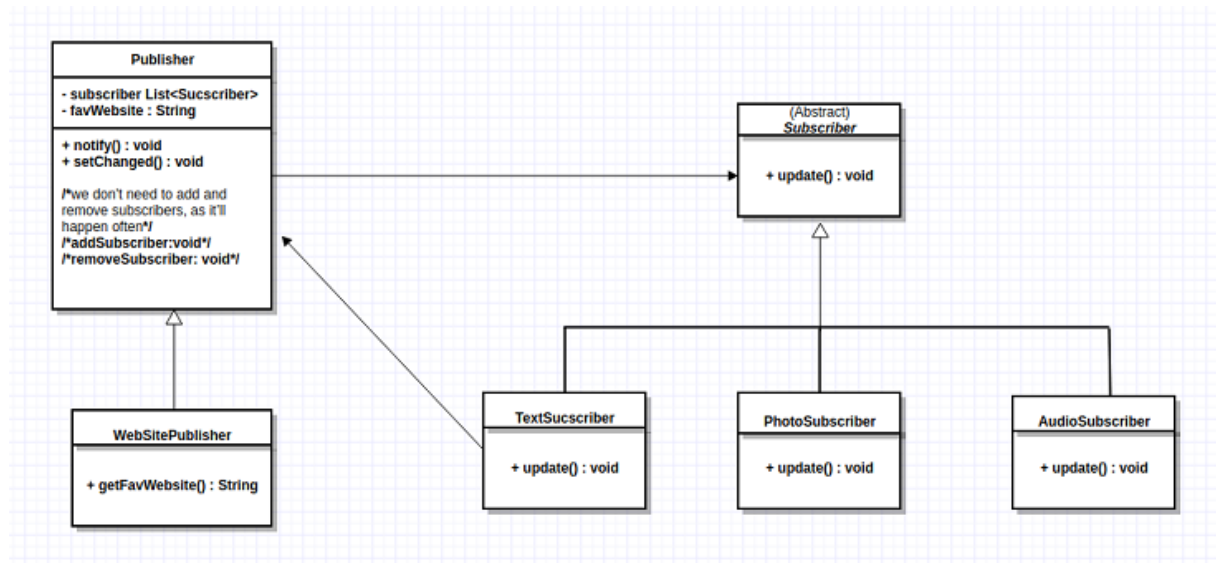
Enter result coefficient

5

6

Gerekli sonuç program tarafından verilir.

Q2. Bu adım için gerekli diagram aşağıda gösterilmiştir. Bu probleme çözüm olarak Observer Pattern ve Decorator Pattern düşünülmüştür. Observer pattern yayıncı ve abone ilişkisi açısından kullanılmıştır. Observer pattern nesneler arasında bir bir bağımlılık ilişkisi tanımlar. Böylece bir nesne durumu değiştiği zaman tüm bağımlı kişiler otomatik olarak güncellenerek bilgilendirilir. (Buradaki websitesi bilgilendirmesi gibi) Aynı zamanda genel olarak abstract class methodları kullanılarak decorator pattern



yönteminden yararlanılmıştır. Bu pattern'e göre yeni bir subscriber veya yeni bir website özelliği gibi yeni içeriklerin eklenebilmesi gibi durumlar için abstract metodu kolaylık sağlamaktadır. Maximum esneklik sağlar.