

PROJEDE İSTENİLENLER:

1. “Imaginary” adında, üye değişkeni (niteliği) ‘private’ olan bir sınıf gerçekleyin. Bu sınıf karmaşık sayıların sanal (Imaginary) kısmını ifade edecektir.
2. Karmaşık sayıları temsil etmek için “Imaginary” sınıfından türettiğiniz (inheritance), “Complex” adında bir sınıf gerçekleyin. Üye değişkenleri, gerçel “re” ve sanal “im”, ve kısımları “private” olmalıdır. a. Bu sınıf üzerinde toplama (+), çıkarma (-), bölme (/), çarpma (*), eşlenik(~), atama (=), toplama atama (+=), çıkarma atama (-=), çarpma atama (*=), bölme atama (/=) ve eşit mi(==) gibi aritmetik ve lojik işlemler aşırı yüklenecektir.
3. Complex sayılar üzerinde tanımlı, “Vector” ve “Matrix” adında sınıfları gerçekleyin. “Vector” sınıfının ‘size_t’ tipinden “size” adında ve “Complex” tipinden “element” adında özellikleri ‘private’ olmalıdır. “Matrix” sınıflara ait ‘size_t’ tipinden “row”, “col” ve “Complex” tipinden “element” adında nitelikleri ‘private’ olmalıdır. Bu sınıflar üzerinde “Complex” sınıfındaki gibi aşağıdaki operatörler aşırı yüklenecektir:

+ (Toplama): operator+	Matrix/Vector nesnesini elemanlarını toplayan operatör
- (Çıkarma): operator-	Matrix/Vector nesnesini elemanlarını çıkaran operatör
* (Çarpma): operator*	Eğer Vector nesnesi ise iç çarpım (dot product), Matrix nesnesi ise matris çarpımı gerçekleştiren operatör
= (Atama): operator=	Bir Matrix/Vector nesnesini başka bir Matrix/Vector nesnesine atayan operatör
+= (Toplama atama): operator+=	Matrix/Vector nesnesini elemanlarını toplayıp atama yapan operatör
-= (Çıkarma atama): operator-=	Matrix/Vector nesnesini elemanlarını çıkarıp atama yapan operatör
= (Çarpma atama): operator=	Eğer Vector nesnesi ise iç çarpım (dot product), Matrix nesnesi ise matris çarpımı gerçekleştiren operatör (bu işlem ancak matrisler kare ise gerçekleştirilebilir)
== (Eşitlik): operator==	İki Matrix/Vector nesnelerinin eşit olup olmadıklarını kontrol eden operatör

- a. “Matrix” sınıfının, matrisin transpozunu hesaplayan “transpose” ve determinantı hesaplayan “determinant” metotlarını gerçekleyin.

4. “Matrix” sınıfından türetilecek “SquareMatrix”, “IdentityMatrix” ve “TriangleMatrix” sınıflarını gerçekleyin.

5. Son olarak, “LinearAlgebraObject” adında soyut (abstract) sınıf gerçekleyin. Bu sınıf, lineer cebir nesnelerinin (örneğin, Vector, Matrix, SquareMatrix, IdentityMatrix, TriangleMatrix) ortak davranışlarını tanımlamak için kullanılacaktır.

Her sınıfa ait “print()”, kurucu (constructor), kopyalama kurucularını (copy constructor) ve gerekli tüm metotlar gerektiği kadar gerçekleştirilmelidir. Tasarladığınız sınıfların testlerini “main” metodunda yapınız ve UML diagramını oluşturunuz.

PROJEDE ALDIĞIM ÇIKTILAR:

-COMPLEX SINIFI ÜZERİNDE YAPTIĞIM İŞLEMLER

```
Complex Number 1: 2 + 3i
Complex Number 2: 1 + -2i
TOPLAM: 3 + 1i
FARK: 1 + 5i
CARPIM: 8 + -1i
BOLUM: -0.8 + 1.4i
COMPLEX1 ESLENIGI: 2 + -3i
Compound Sum: 3 + 1i
Compound Difference: 1 + 5i
Compound Product: 8 + -1i
Compound Quotient: -0.8 + 1.4i
Complex sayilar esit mi: false
*****
```

-VECTOR SINIFI ÜZERİNDE YAPTIĞIM İŞLEMLER

```
Vector 1:[1 + 2i, 3 + 4i, 5 + 6i]
Vector 2:[7 + 8i, 9 + 10i, 11 + 12i]
TOPLAM: [8 + 10i, 12 + 14i, 16 + 18i]
FARK:[-6 + -6i, -6 + -6i, -6 + -6i]
CARPIM:[-9 + 22i, -13 + 66i, -17 + 126i]
Compound sum: [8 + 10i, 12 + 14i, 16 + 18i]
Compound difference: [1 + 2i, 3 + 4i, 5 + 6i]
Compund product: [-9 + 22i, -13 + 66i, -17 + 126i]
Vektorler esit mi?: false
*****
```

-MATRİX SINIFI VE MATRİX SINIFINDAN TÜRETTİĞİMİZ CLASSLAR ÜZERİNDEKİ İŞLEMLER:

```
Matrix 1:
1 + 2i 3 + 4i
5 + 6i 7 + 8i

Matrix 2:
9 + 10i 11 + 12i
13 + 14i 15 + 16i

TOPLAM:
10 + 12i 14 + 16i
18 + 20i 22 + 24i

FARK:
-8 + -8i -8 + -8i
-8 + -8i -8 + -8i

CARPIM:
-28 + 122i -32 + 142i
-36 + 306i -40 + 358i

Compound sum:
10 + 12i 14 + 16i
18 + 20i 22 + 24i

Compound difference:
1 + 2i 3 + 4i
5 + 6i 7 + 8i

Compund product:
-28 + 122i -32 + 142i
-36 + 306i -40 + 358i
```

```
Matrix 2'in transpozu:
9+10i   13+14i
11+12i   15+16i

Matrix 1'in determinanti: -32

Matrixler esit mi? false
*****
Square Matrix:
1 + 2i 3 + 4i
5 + 6i 7 + 8i

Triangle Matrix:
1 + 0i 2 + 0i 3 + 0i
0 + 0i 4 + 0i 5 + 0i
0 + 0i 0 + 0i 6 + 0i

Identity Matrix:
1 + 0i 0 + 0i
0 + 0i 1 + 0i
```

PROJEDE ÖĞRENDİKLERİM VE ZORLANDIĞIM KISIMLAR:

Projedeki son kısım olan abstract sınıf oluşturma kısmını tam anlayamadım anladığım kadarıyla bişeyler ekledim kodum sıkıntı çıkmadan çalıştı. Onun dışında for döngülerinin algoritmasını anlamakta zorluk çektim matrix sınıfı için reel ve imajiner kısmı için ayrı ayrı işlemler yaparken zorlandım. Onun dışında c++ dilini daha iyi anladığımı düşünüyorum. C++'ta kalıtımı, constructorları, copy constructorları ve kullanımını daha iyi anladım. "+" ve "+=" operatörleri arasındaki farkı daha iyi anladım.

PROJEDEKİ EKSİKLİKLER:

Abstract "LinearAlgebraObject" sınıfını oluşturdum fakat çalışma mantığını tam anlamadım diyemem. Tekrar eden fonksiyonları ekledim. İnternette biraz baktım fakat projeme uyarlayamadım. Diğer hatam ise, determinant hesaplama kısmında sanal kısmı hesaplamaya katmıyor matrixteki reel sayılarla işlemi yapıyor ve bu işlemi sadece 2 boyutlu matrixler için yapıyor. Girilen boyuta göre determinant hesaplamasını yapamadım. Onun dışında projemin çıktılarını raporuma ekledim.

PROJEMİN UML DİAGRAMI:

