

BLM19204-BLM204 BILGISAYAR PROGRAMLAMA III DERSİ 2. PROJE RAPORU

PROJEDE İSTENİLENLER

Bu projede, C++ programlama dili kullanılarak, karmaşık sayılar uzayında tanımlı olan vektör ve matris üzerinde bir takım lineer cebir işlemleri gerçekleştiren sınıf ve metotları gerçeklemeniz ve main metodu içerisinde test etmeniz beklenmektedir.

Gerçeklenmesi gereken sınıflar ve ilgili metotları aşağıda verilmiştir:

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

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

4. “Matrix” sınıfından türetilen “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.

PROJEDEN ÖĞRENDİKLERİM

Bu projede, C++ dilinde oluşturduğumuz classlara operatör ve metotlarla istenilen lineer cebir işlemlerini gerçeklemeyi öğrendim. Projede istendiği üzere çeşitli class ve metotlar için türetme (inheritance) işlemi yapmayı ve gerçeklediğim soyut sınıfta oluşturduğum lineer cebir nesnelerinin ortak davranışlarını tanımladım.

PROJEDE YAPTIKLARIM

Bu projede, “Imaginary” adında, üye değişkeni (niteliği) ‘private’ olan bir sınıf gerçekledim. Bu sınıf karmaşık sayıların sanal (Imaginary) kısmını ifade ediyor.

Karmaşık sayıları temsil etmek için “Imaginary” sınıfından türettiğim (inheritance), “Complex” adında bir sınıf gerçekledim. 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 bulunuyor.

Complex sayılar üzerinde tanımlı, “Vector” ve “Matrix” adında sınıfları gerçekledim. Bu sınıflarda; + (Toplama): operator+, - (Çıkarma): operator-, *(Çarpma): operator*, = (Atama): operator=, += (Toplama atama): operator+=, -= (Çıkarma atama): operator-=, *= (Çarpma atama): operator*=, == (Eşitlik): operator== ve “Matrix” sınıfının, matrisin transpozunu hesaplayan “transpose” ve determinantı hesaplayan “determinant” metotları bulunuyor. “Matrix” sınıfından türettiğim “SquareMatrix”, “IdentityMatrix” ve “TriangleMatrix” sınıflarında özel matrixleri oluşturan creat metotları bulunuyor.

Son olarak, “LinearAlgebraObject” adında soyut (abstract) sınıf gerçekledim. Bu sınıf, lineer cebir nesnelerinin ortak davranışlarını tanımlamak için kullandım.

Tasarladığım sınıfların testlerini “main” metodunda yaptım.

PROJEDE EKSİK BIRAKTIĞIM VE ZORLANDIĞIM KISIMLAR

Bu projede zorlandığım kısımlar, vector ve matrix sınıfları için oluşturduğum atama operatörlerinde (+=, -=, *=) ne istendiğini anlamak, bu operatörleri gerçeklemek, doğru sonuçları almak, matrix sınıfında oluşturduğum determinant hesaplayan determinant metodundan doğru sonucu almak, LinearAlgebraObject adındaki abstract sınıfı oluşturmak.

PROJE ÇIKTILARIM

```
1.Complex: 3 + 2i
2.Complex: 5 + 4i

Sum (+): 8 + 6i
Difference (-): -2 + -2i
Product (*): 7 + 22i
Quotient (/): 0.560976 + 0.0487805i
Conjugate of c1: 3 + -2i
Equality(==): c1 is not equal to c2

Sum (+=): 8 + 6i
Difference (-=): -2 + -2i
Product (*=): 7 + 22i
Quotient (/=): 0.560976 + 0.0487805i

1.Vector:
1 + 2i 3 + 4i 5 + 6i

2.Vector:
7 + 8i 9 + 10i 11 + 12i

Sum of vectors:
8 + 10i 12 + 14i 16 + 18i

Difference of vectors:
-6 + -6i -6 + -6i -6 + -6i

Multiplication of vectors:
7 + 22i

Sum Vector(+=):
8 + 10i 12 + 14i 16 + 18i

Difference Vector(-=):
-6 + -6i -6 + -6i -6 + -6i

Product Vector(*=):
-9 + 22i -13 + 66i -17 + 126i

Vectors are not equal.
```

```
1.Matrix:
4 + 1i 6 + 3i
10 + 7i 9 + 10i

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

Determinant of first matrix:
-13 + -23i

Transpose of first matrix:
4 + 1i 10 + 7i
6 + 3i 9 + 10i

Sum of matrices:
5 + 3i 9 + 7i
15 + 13i 16 + 18i

Difference of matrices:
3 + -1i 3 + -1i
5 + 1i 2 + 2i

Multiplication of matrices:
14 + 60i 26 + 88i
-19 + 131i -15 + 203i

Sum Matrix(+=):
5 + 3i 9 + 7i
15 + 13i 16 + 18i

Difference Matrix(-=):
3 + -1i 3 + -1i
5 + 1i 2 + 2i

Product Matrix(*=):
14 + 60i 26 + 88i
-19 + 131i -15 + 203i

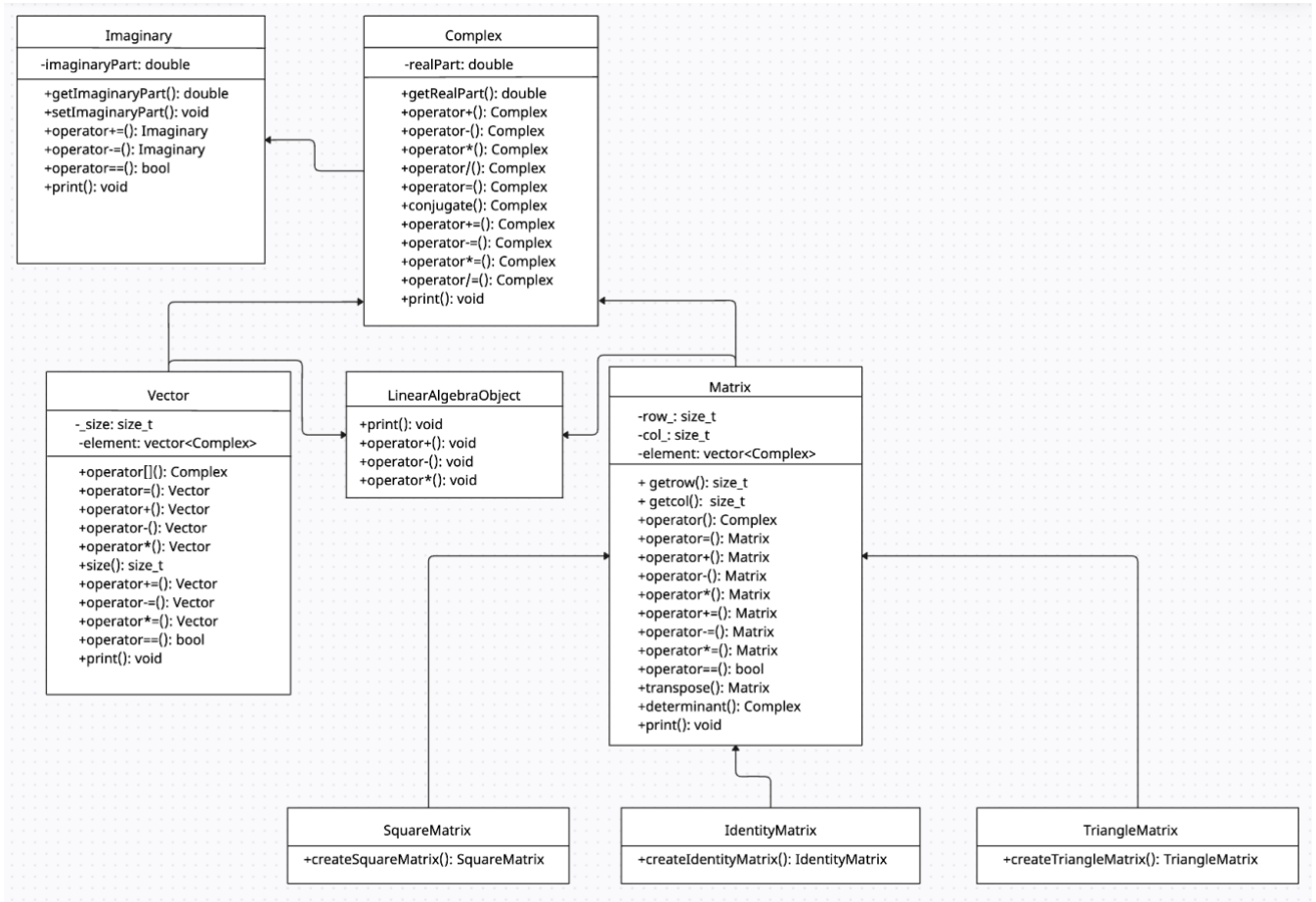
Matrices are not equal.
```

```
Square Matrix:  
4 + 1i 6 + 3i  
10 + 7i 9 + 10i
```

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

```
Triangle Matrix:  
4 + 1i 0 + 0i 0 + 0i  
6 + 3i 10 + 7i 0 + 0i  
2 + 5i 8 + 9i 1 + 3i
```

UNIFIED MODELING LANGUAGE (UML) DİYAGRAMI



Hacer Ebra ÖZDEMİR
2121221045
Bilgisayar Mühendisliği