

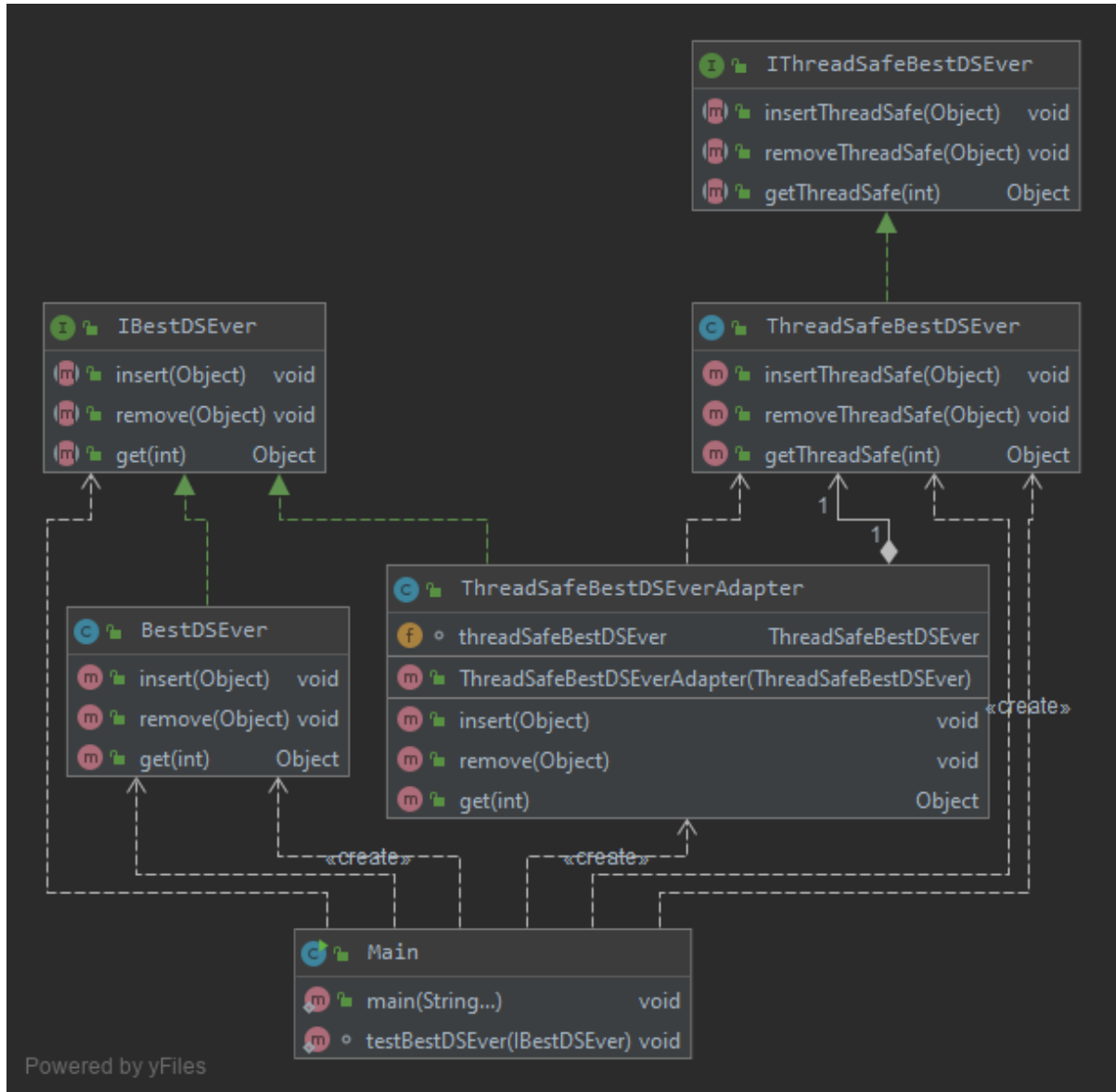
Gebze Technical University
Computer Engineering

Object Oriented Analysis and Design
CSE443 – 2019

HOMEWORK 3 REPORT

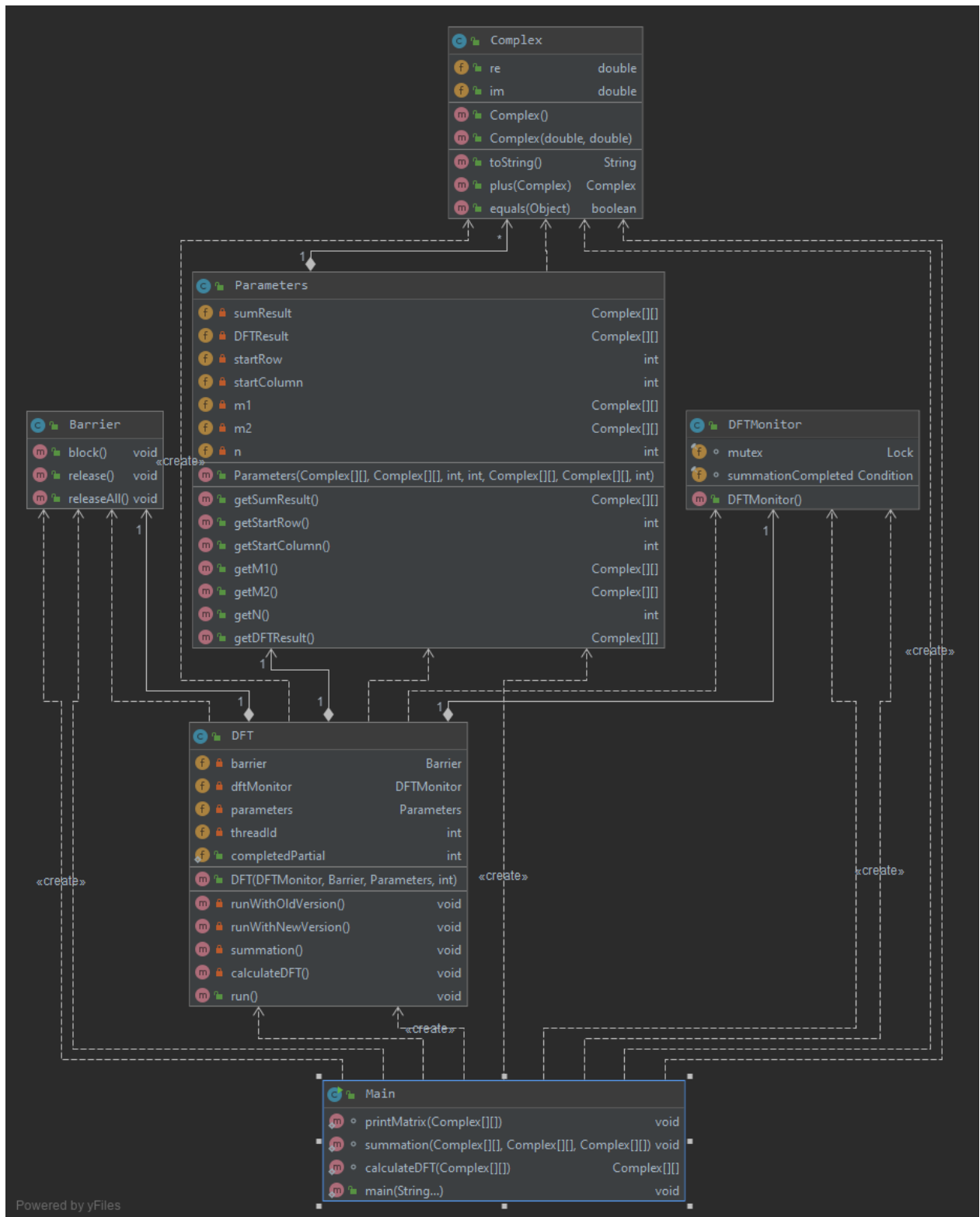
Serkan Sorman
151044057

Q1)



Probleme çözüm olarak Adapter design patternını uyguladım. Çünkü hem iki data structure in interfacesi farklı hemde bu iki yapının davranışları neredeyse aynı. ThreadSafe olarak tanımlanan yapıda her method için synchronized keywordu eklendi. Böylece eş zamanlı olarak bu veri yapısına erişim tek bir thread ile sınırlandırılmış oldu. Bir adapter sınıfı yazıldı ve içinde thread safe olan veri yapısı tutuldu. Thread safe olmayan yapının methodları implement edilerek thread safe olarak güvenli olan yapının metodları implement edilen metodlar içerisinde kullanıldı.

Q2)

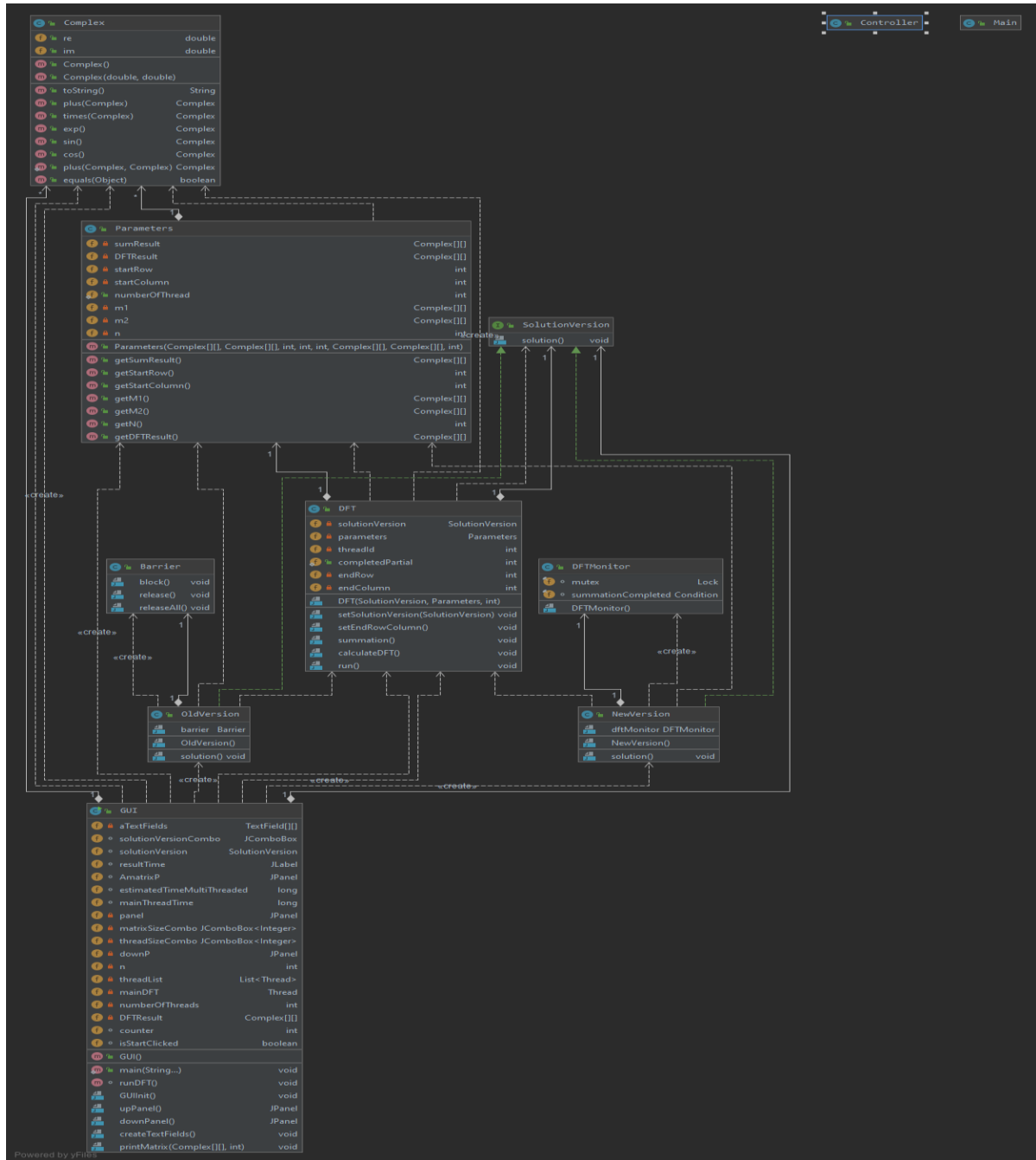


Synchronization barrier problemine çözüm olarak ilk yöntemde bir Barrier classı oluşturuldu ve içerisinde wait, notify, notifyAll metodları implement edildi. Her thread matrisler üzerinde kendi bölümünü topladıktan sonra wait ile bekletildi. Ardından toplama işlemini tamamlayan thread sayısı 4 olana kadar diğer threadler de aynı şekilde bekletildi. Son thread

toplama işlemini bitirdikten sonra diğer tüm diğerleri notifyAll ile uyandırdı. Böylece tüm threadler toplama işlemi tamamlandıktan sonra aynı mantıkla kendi bölgelerindeki DFT'yi hesapladılar.

İkinci yöntemde ise bir monitör sınıfı oluşturularak içerisinde Lock ve Condition yapıları tutuldu. Toplama işlemi tamamlayan threadler mutexi kilitledi ve bu işlemi tamamlayan thread sayısı 4 olma conditionu ile while içerisinde await edildi. Son thread toplama işlemini tamamladı ve bu whilea girmeden diğer tüm threadleri condition variable üzerinden signalAll ile uyandırdı. Ardından tüm threadler unlock edildi. Böylece tüm threadler toplama işlemini tamamladıktan sonra DFT hesaplamasına geçiş yaptılar.

Q2 GUI)



GUI ekranında kullanıcıya belirli matris boyutları seçme hakkı sunuldu. Ayrıca kullanıcı isteğe göre 1,2,4 ve 8 thread kullanarak DFT hesaplaması gerçekleştirebiliyor. GUI üzerinden seçilen oldversion ile wait ve notifyAll metodları kullanılarak yapılan senkronizasyon yöntemi seçilmiş olur. NewVersion seçildiğinde ise monitör kullanılarak yapılan senkronizasyon yöntemi seçilmiş olur. Kullanıcı başlattığı herhangi bir hesaplamayı cancel butonuna basarak durdurup iptal edebilir ve tekrar bir hesaplama başlatabilir. Ayrıca kullanıcı herhangi bir hesaplama yaptıktan sonra “Compare with Single thread” butonuna basarak single thread ile çalıştırmaya göre kazandığı zamanı görüntüleyebilir. Kullanıcı hiçbir hesaplama yapmadan compare butonuna bastığında önce bir hesaplama yapması gerektiği ile ilgili uyarıyla karşılaşır.