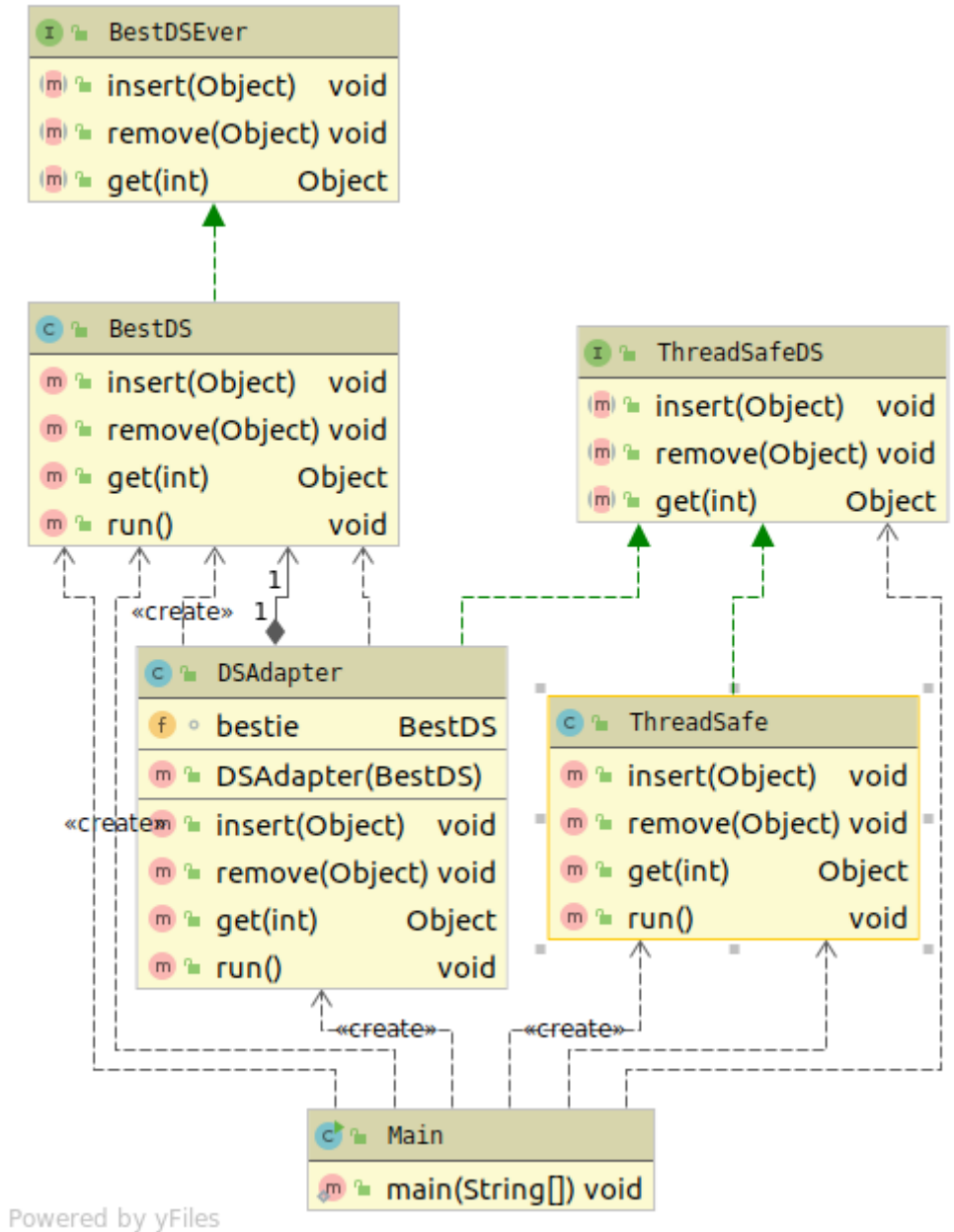
OBJECT ORIENTED ANALYSIS AND DESIGN
HW03 RAPORU

Emire Korkmaz
141044043

Question 1)

Sorunun çözümü için Adapter Tasarım Örüntüsü kullanılmıştır. Bu şekilde BestDSEver sınıfında herhangi bir değişiklik yapılmadan thread safe hale getirilmiştir. İki tane interface oluşturulmuştur; BestDSEver ve ThreadSafeDS'tir. Bunları implement eden BestDS, ThreadSafe ve DSAdapter sınıfları oluşturulmuştur. Sınıf şeması alttaki gibidir.



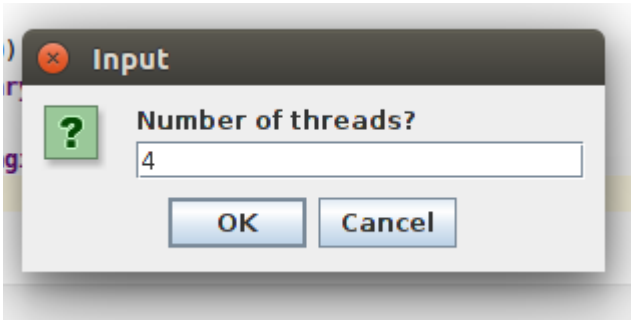
Question 2)

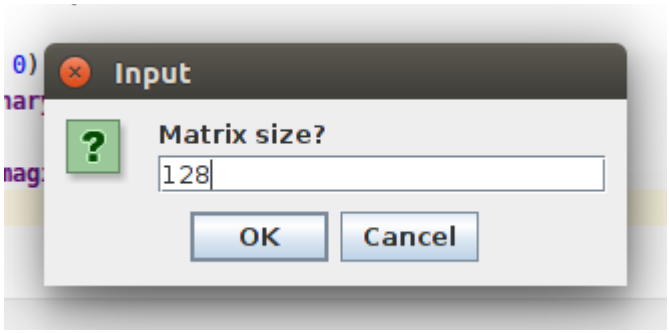
Program çalıştırılınca kullanıcıdan Thread sayısını, matris boyutunu ve DFT bulunurken kullanılacak olan senkronizasyon yöntemini alır.

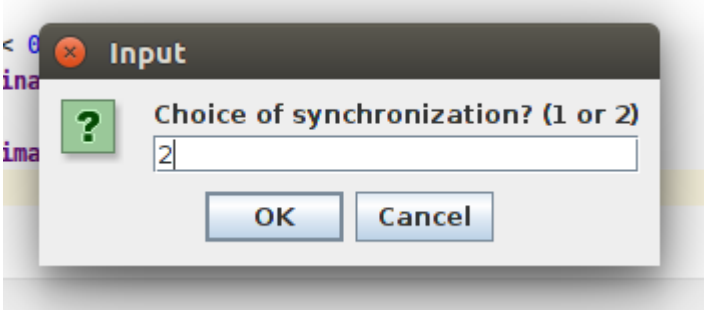
→ Senkronizasyon yöntemi olarak sunulan seçenekler 1 ve 2'dir.
- 1, **wait(), notify()** ve **notifyAll()** fonksiyonları kullanılarak **DFT** hesaplayan yöntemdir.
- 2 ise **mutex** ve **monitor** vb. kullanan yöntemdir.

→ Hesaplamalar yapılırken başka bir pencere daha açılır. Bu pencere, kullanıcıya işlemleri iptal etme ve tekrardan başlatma imkanı sağlar.
- **CANCEL** butonuna tıklandığında var olan thread'ler durdurulur ve **mutex**'ler relase edilir.
- **RESTART** butonuna tıklandığında tekrardan thread sayısı, matris boyutu ve senkronizasyon yöntemi alınarak en baştan hesaplama yapılır.

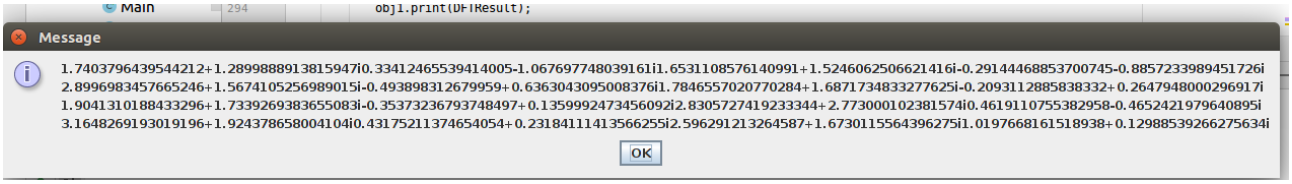
Kullanıcının işlemi iptal etmediği durum için örnek görseller;



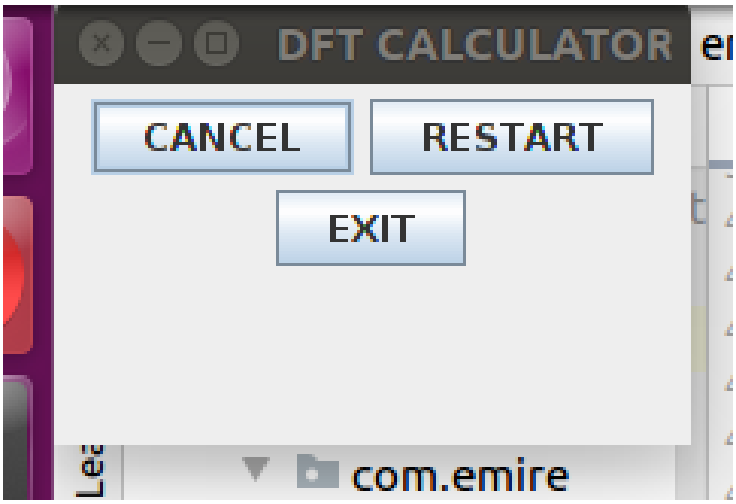


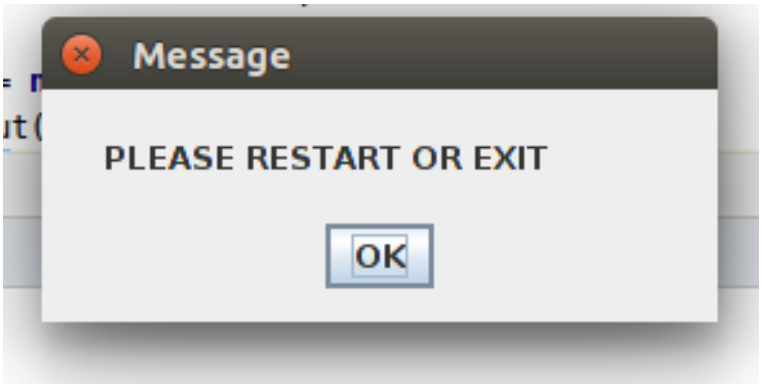


Sonuç terminal ekranının yanı sıra GUI olarak da kullanıcıyla paylaşılır.

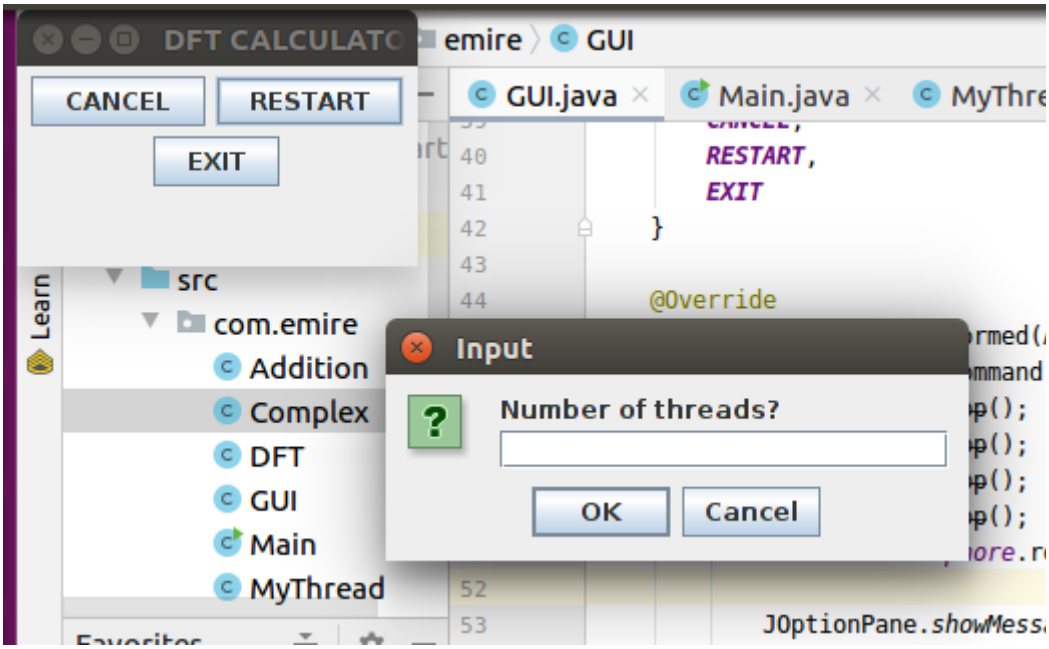


Diğer bir durumda ise aynı işlemler tekrarlandıktan sonra kullanıcı işlemi iptal etmek için **CANCEL** butonuna bastığında işlemler durur ve kullanıcının tekrardan **RESTART** butonuna basması istenilir.

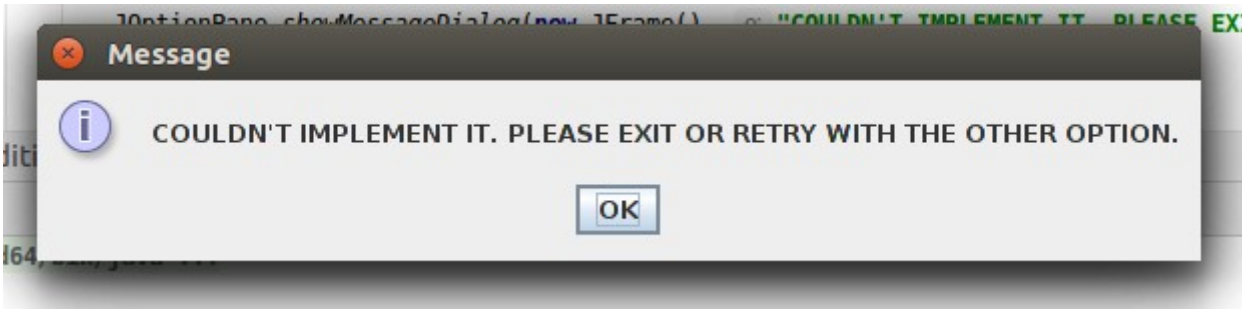




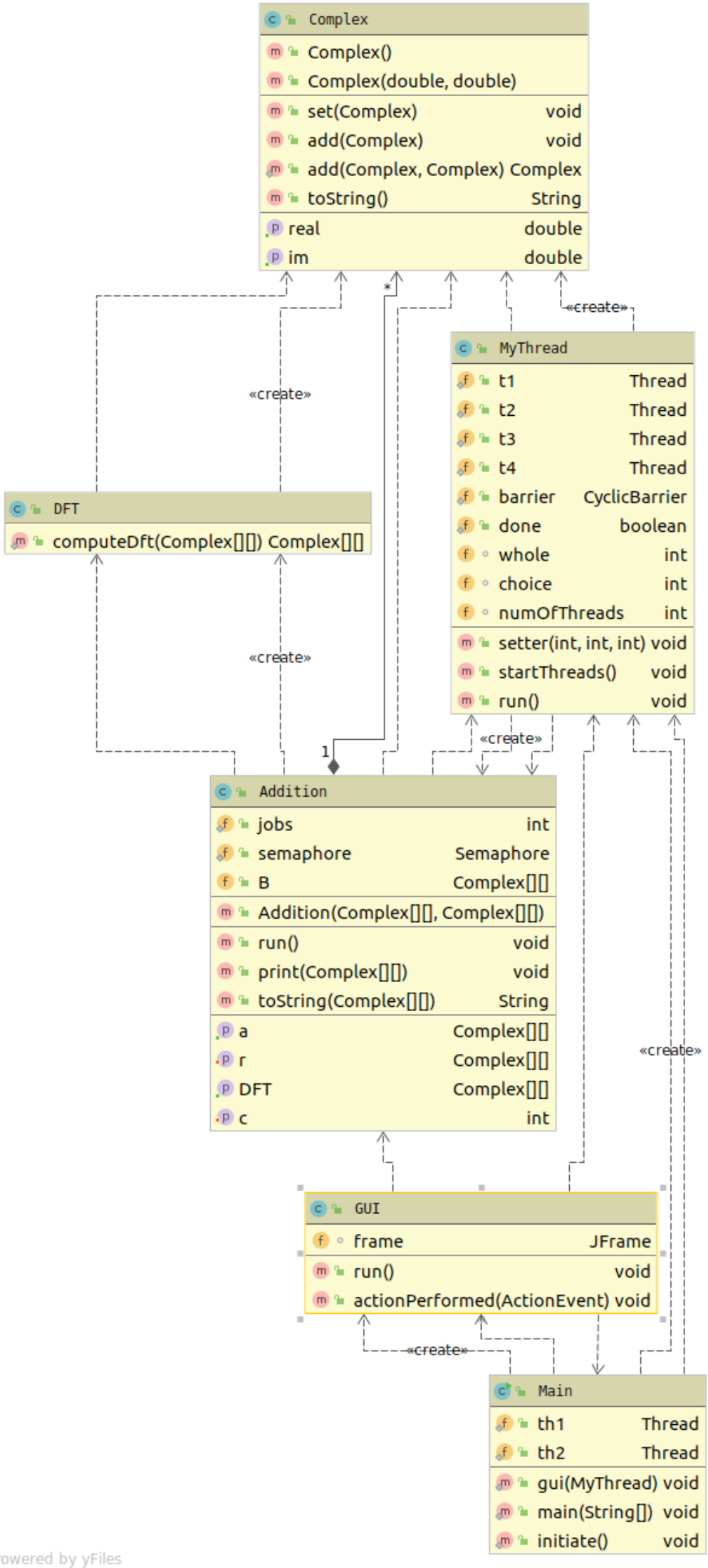
RESTART'a basıldıktan sonra tekrardan başlangıç ekranı gelir.



NOT: İlk seçenek olan wait(), notify ve notifyAll() kullanarak yapılan hesaplamayı yapamadım. Sadece mutex ve monitor olan seçenek mevcuttur. İlk seçenek seçildiğinde alttaki mesaj belirmektedir.



→ İkinci kısmın sınıf şeması aşağıdaki gibidir.


Complex
Complex()
Complex(double, double)
set(Complex) void
add(Complex) void
add(Complex, Complex) Complex
toString() String
real double
im double
«create»
*
MyThread
t1 Thread
t2 Thread
t3 Thread
t4 Thread
barrier CyclicBarrier
done boolean
whole int
choice int
numOfThreads int
setter(int, int, int) void
startThreads() void
run() void
DFT
computeDft(Complex[][]) Complex[][]
1
Addition
jobs int
semaphore Semaphore
B Complex[][]
Addition(Complex[][], Complex[][])
run() void
print(Complex[][]) void
toString(Complex[][]) String
a Complex[][]
r Complex[][]
DFT Complex[][]
c int
GUI
frame JFrame
run() void
actionPerformed(ActionEvent) void
Main
th1 Thread
th2 Thread
gui(MyThread) void
main(String[]) void
initiate() void
Powered by yFiles