

CSE 443 OBJECT ORIENTED ANALYSIS AND DESIGN

ÖDEV 3- RAPOR

AKIN ÇAM
151044007

ÖDEV 3- RAPOR

2019-2020 GÜZ DÖNEMİ KAPSAMINDA ÖDEV-3 RAPORU

TANIM VE AÇIKLAMALAR

IntelliJ İdea ortamında geliştirilmiştir. 3 package halinde geliştirilmiştir.

Birinci package BestDSEver, ikinci package DFT' gerçekleştirilmesi, üçüncü package Arayüz olarak belirlenmiş ve gerçekleştirilmiştir.

BÖLÜM 1:

1. BestDSEver ThreadSafe gerçekleştirirken Adapter sınıfı structural tasarım desenlerinden biridir.

2. Bu tasarım deseni, birbiriyle ilişkili olmayan interface'lerin birlikte çalışmasını sağlar. Bu işlemi ise, bir sınıfın interface'ini diğer bir interface'e dönüştürerek yapar.

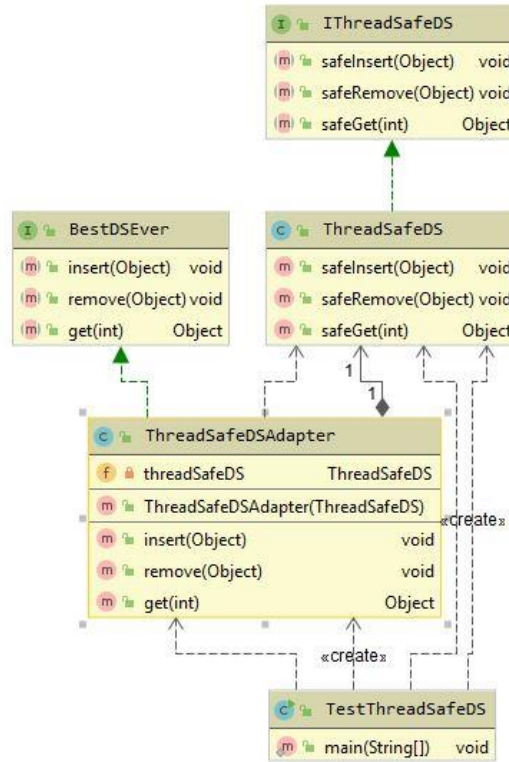
Bu bilgiler ışığında BESTDSEver sınıfını thread safe hale getirmek için Adapter sınıfı kullanılmıştır.

Bir ThreadSafe arayüz sınıfı oluşturulmuştur.

Bu sınıftan bir ThreadSafeDSEver gerçekleştirilmiş ve methodlar thread safe olarak geliştirilmiştir.

Daha bu arayüzleri ilişkilendirmek için bir Adapter Sınıfı oluşturulmuş ve bu sınıfta BestDSEver arayüzü gerçekleştirilmiştir. Bu sınıf içinde threadSafe olan ThreadSafeDSEver değişkeni tutulmaktadır. Böylece arayüzlerin birbiriyle çalışması sağlanmıştır.

Sınıf Diagramı:



Sınıf diagramından da görüldüğü gibi İki ayrı interface Adapter sınıfı ile dönüştürülmesi sağlanmıştır. ThreadSafeAdapter sınıfı ThreadSafeDS sınıfını bulundurmaktadır ve BESTDSEver gerçekleştirilmiştir.

----->
ThreadSafeDS Adapter----->

Thread Safe Insertion

Thread Safe Remove

Thread Safe Get

Process finished with exit code 0

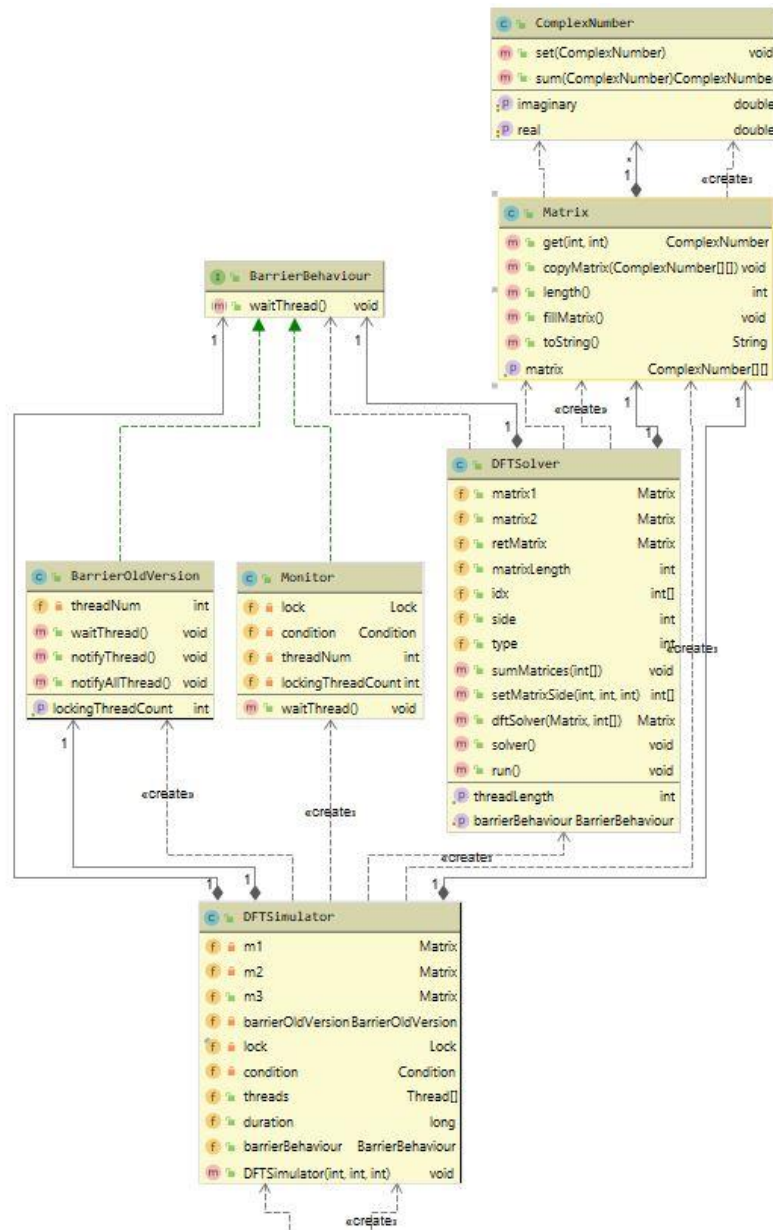
BÖLÜM 2:

Bu bölümde Barrier Senkronizasyon methodları ve Strategy Tasarım Deseni kullanılmıştır. Strategy Tasarım Deseni Barrier Sınıfları gerçekleştirirken kullanılmıştır. Çünkü iki farklı barrier davranışı mevcuttur. Bir BarrierBehaviour arayüzü gerçekleştirilmiş ve içine barrier methodu boş olarak bırakılmıştır. Bu sınıf OldversionB ve Monitor Sınıfını gerçeklemektedir. OldVersionB sınıfında bir thread sayısı tutulur thread sayısı tamamlanmadığı sürece wait durumunda tüm threadler bekletilir. Son thread ulaştığında notifyAll ile sonuca ulaşılır. Monitor sınıfında Lock ve Condition sınıfları kullanılmıştır. Methoda giren thread bekletilir (while döngüsü) son thread

ulaştığında tüm threadlere sinyal gönderilmektedir Dft fonksiyonu çözümü için şu adımlar izlenmiştir: İlk olarak thread size ve matris size belirlenir. Daha sonra threadler bölünür ve matrisler toplanır. Toplam olan matris DFT ye gönderilir ve sonuç hesaplanır. Bu DFTSolver sınıfı runnable olarak gerçekleştirilmiştir. Thread sayısı kadar bu sınıf oluşturulmuş ve çalıştırılmıştır. Bu sınıfta bulunan BarrierBehaviour fonksiyonuna göre Barrier fonksiyonu seçilir. Karmaşık sayılar için bir ComplexNumber sınıfı, Bunu bir matrisi için matrix sınıfı gerçekleştirilmiştir. DFT matris te şu formül kullanılmıştır:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk / N}$$

Sınıf Diagramı:



BÖLÜM 3:

Bu bölümde MVC yapısına uygun olarak arayüz gerçekleştirilmiştir. Bir View Sınıfında Arayüz oluşturulmuştur. Model sınıfında DFT methodu bir thread içinde çağrılmaktadır. Çünkü fonksiyon çalışırken arayüz kontrol edilebilir olmalıdır. Controller sınıfında başla ve bitir butonlarının davranışları gerçekleştirilmiştir. Start da DFT methodu ayrı bir thread ile ekrandaki text leri düzenleyen method ayrı bir thread ile çağrılmış. Stop durumunda tüm thread ler DFT methodunu çözen de olmak üzere durdurulmuştur. DFT methodunu çözen thread işini bitirdiği kontrol edilir ve textler güncellenir.

Sınıf Diagramı:

