

PDF te belirtildiği üzere her iki bölüm için de proxy pattern tercih edildi. DatabaseTable sınıfını temsil etmek amacıyla bu sınıf ITable interface'den implement edildi bu sınıftaki metodlar çağırıldıklarında gerçek sınıftan bir metodun çağırıldığını anlamak amacıyla real tagi ile print edildi.

GetElementAt bir read çağrısı ve SetElementAt ise aslında bir write çağrısıdır.

Paralel şekilde read işleminde herhangi bir sorun ortaya çıkmayacaktır ancak write işlemi yapılırken hiçbir thread ne write işlemi ne de read işlemi yapamamalıdır. Bu nedenle 4a bölümünde readwrite lock kullanılmıştır. Javanın

ReentrantReadWriteLock sınıfı paralel şekilde read işleminin yapılabilmesini ve write işlemi yapılırken read ve write işleminin yapılamayacağını garanti etmektedir ancak bu sınıf write işlemi için bir öncelik sağlamaz.

4b bölümü için write işlemi yapan threadlere öncelik vermeliyiz ki çok fazla read işlemi yapmak isteyen thread olduğunda açlık durumuna girmesin.

Javada read write lock önceliğini ayarlayabileceğim bir sınıf bulamadığımdan PrioWriteLock sınıfı implement ettim.

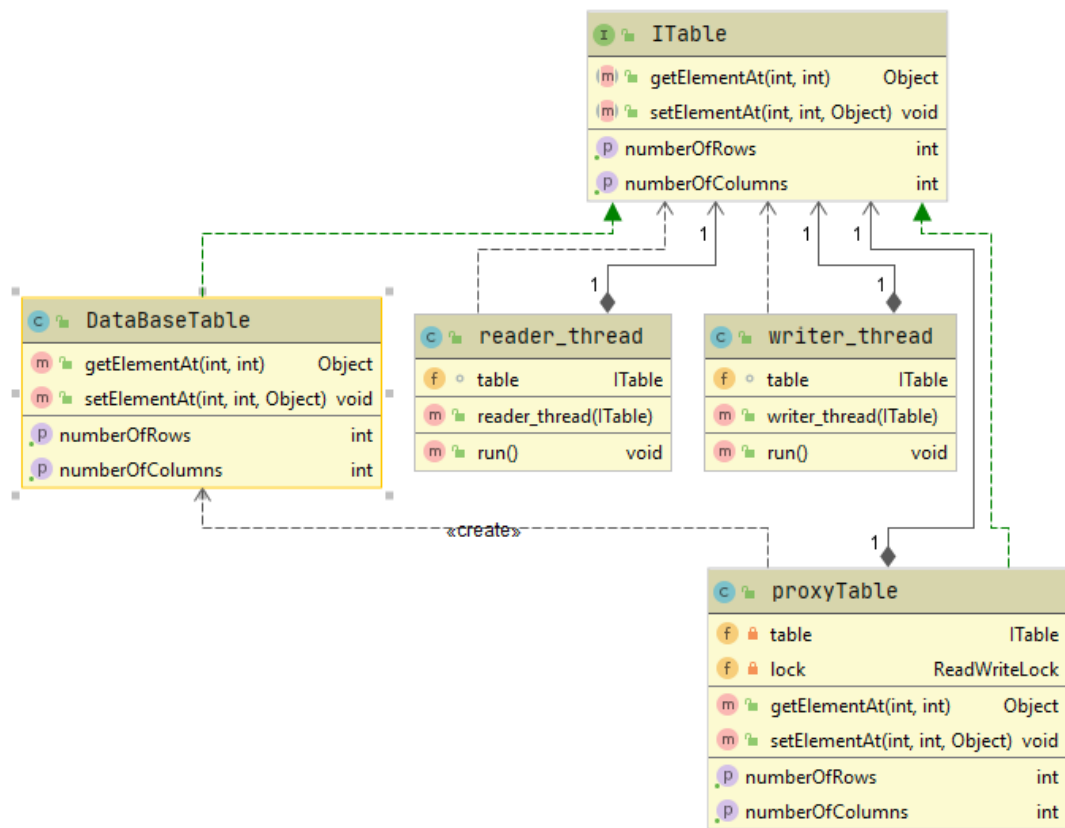
Eğer read threadleri yalnızca o anda herhangi bir write işlemi yapılmıyorsa ve write işlemi yapmak isteyen bir writer thread write request etmediyse işlemini yapabilir. Diğer tüm durumlarda beklemek zorundadır. Diğer özellikler 4a bölümünde kullandığım read write lock ile aynıdır.

Proxy tasarım dizaynını **proxyTable** ve **ProxyWithPriority** sınıfları ile gerçekledim.

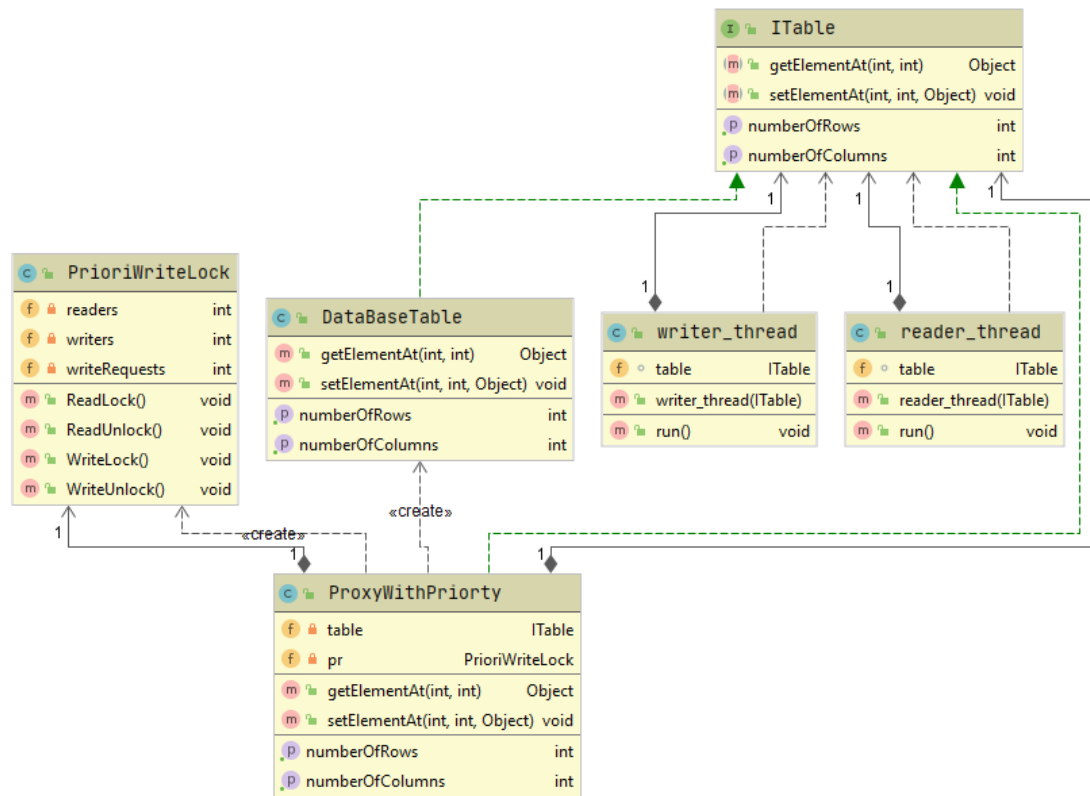
proxyTable sınıfı 4a bölümü için ReentrantReadWriteLock kullanıyor.

ProxyWithPriority sınıfı ise 4b bölümü için implement ettiğim PrioWriteLock kullanıyor.

4a class diagram :



4b class diagram:



Output:

Çalıştırmak için herhangi bir input gereksinimi olduğunu düşünmediğimden sırasıyla 1 adet read thread ve 1 adet write thread olmak üzere 10 ar adet read ve write threadlerini bir arraylist'e ekledim ve 20 threadin tamamını çalıştırdım.

4a bölümü için output'u incelersek :

Proxy sınıfı get ve set metodu çağırıldığında sırasıyla start get/set daha sonra gerçek sınıfın read tagli kendi print işlemi ve end get/set printini basıyor.

Bu printlerin block olarak basılması işlemin bölünmeden tamamlandığı anlamına geliyor.

Örneğin outputta görüldüğü üzere get işlemi paralel çalıştığı için get blokları bazen iç içe girmiş ancak hiçbir set block bölünmemiş bu write işleminin bölünmeden tamamlandığını ve bu sırada herhangi bir get veya set işleminin yapılmadığını gösteriyor.

```
PS C:\Users\YERMA\IdeaProjects\hw2_part4\out\artifacts\hw2_part4_jar> java -jar .\hw2_part4.jar
Sırasıyla birer birer 10'ar adet read ve write threadleri çalıştırılıyor

start proxy set
real set element
end proxy set

start proxy set
real set element
end proxy set

start proxy set
real set element
end proxy set

start proxy get
real get element
end proxy get

start proxy set
real set element
end proxy set

start proxy get
real get element
end proxy get

start proxy get
start proxy get
start proxy get
real get element
end proxy get

start proxy get
real get element
end proxy get

real get element
start proxy get
real get element
end proxy get

start proxy get
real get element
end proxy get

end proxy get

start proxy set
real set element
end proxy set

start proxy get
real get element
start proxy get
real get element
end proxy get

end proxy get

start proxy set
real set element
end proxy set

start proxy set
real set element
end proxy set

start proxy set
real set element
end proxy set
```

⇒ parallel read

4b bölümü için output'u incelersek:

Resimde görüldüğü üzere write işlemi öncelikli olduğu için threadler aynı anda çalıştırılmasına rağmen set işlemleri önceliklendirilmiş ve get işlemlerinden önce yapılmış ve yine burada görüldüğü üzere set işlemleri block olarak bölünmemiş get işlemleri ise paralel çalışabildikleri için blocklar birbirine girmiş durumda

Write threadleri öncelikli olacak şekilde implement edilen proxy pattern
Sırasıyla birer birer 10'ar adet read ve write threadleri çalıştırılıyor

```
start proxy get  
real get element  
end proxy get
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy set  
real set element  
end proxy set
```

```
start proxy get  
real get element
```

```
start proxy get  
start proxy get  
start proxy get  
start proxy get  
start proxy get  
start proxy get  
real get element  
start proxy get  
real get element  
start proxy get  
start proxy get  
real get element  
end proxy get
```

```
end proxy get
```

```
real get element  
real get element  
real get element
```