

Gebze Technical University  
Computer Engineering

CSE 222  
2017 Spring

HOMEWORK 3 REPORT

HÜDAİ SEMİH ÇAVDAR  
151044081

## 1. Detailed System Requirements

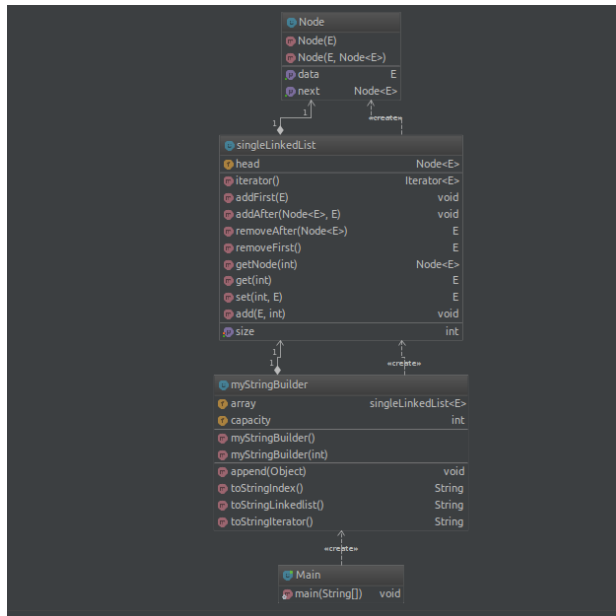
**Q1->** SingleLinkedList classını ve inner classı olan Node classını kitaptan aldım ve generic yapısı ile birleştirerek kullandım. Sonrasında myStringBuilder class'ının içinde SingleLinkedList objesi oluşturdum. Add ve remove işlemlerini yaparken SingleLinkedList objesi içinde ki list'i kullanarak işlem yapıyor.

**Q2->** SingleLinkedList'e önce ekleme yaptım sonrasında recursive olarak yazdığım reverseToString metodunu çağırdım. Sona kadar gidip sondan başa gelerek çalışan metodun döndürdüğü String'i ekrana yazdırarak kontrol ettim.

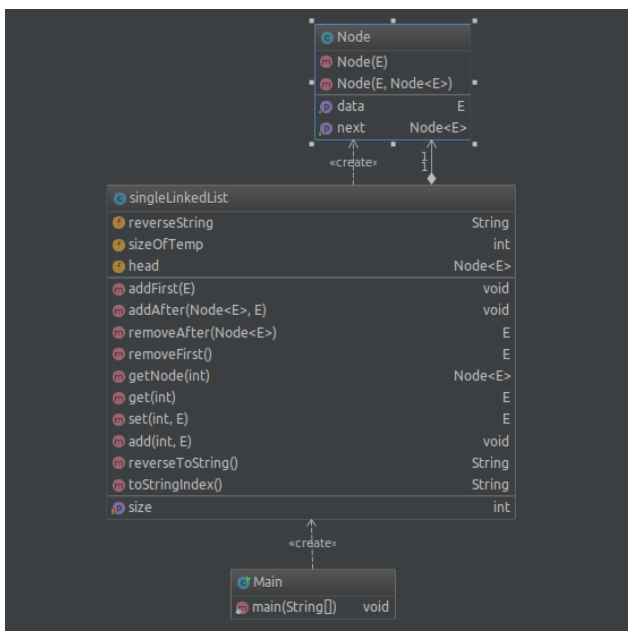
**Q4->** Silinen node'ları bir arraylistte tuttum ve lazım olduğu zaman yeni node oluşturmak yerine silinmiş node'lardan birini kullanıyorum. Bu şekilde garbage collector daha az iş yapıyor ve performans açısından daha iyi bir sonuç alıyoruz.

## 2. Class Diagrams

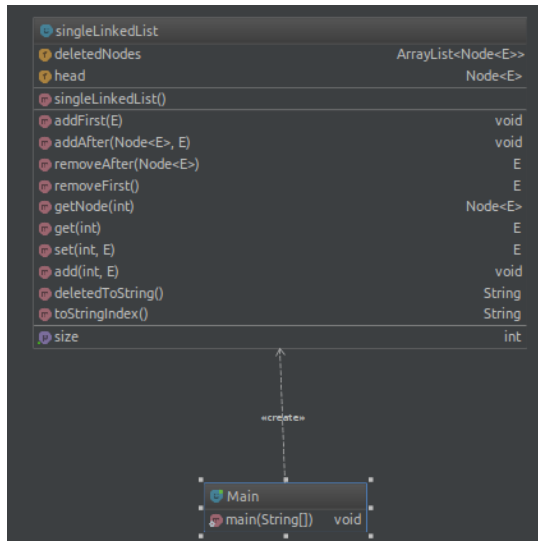
**Q1->**



**Q2->**



Q4->



## 5. Problem Solutions Approach

Q1-> SingleLinkedList implementasyonu için Node<E> classını yazıp **composition** ile SingleLinkedList'in data field'inde list'in başını göstermek için "head" isimli bir Node<E> objesi oluşturmamız gerekir.

MyStringBuilder class'ı için ise **composition** kullanarak SingleLinkedList objesi oluşturmamız gerekir.

Q2-> SingleLinkedList'i burda da kullandığım için Node<E> classının objesini **composition** kullanarak SingleLinkedList'in data field'inde oluşturdum.

Q4-> SingleLinkedList'i kullandım ve datafield'inde **composition** yaptım. SingleLinkedList'in içinde silinecek olan Node<E> objelerini saklamak için bir ArrayList oluşturdum. Add işlemi yapıldığında ArrayList'ten "remove" işlemi yaptım. Remove işlemi yapıldığında ise ArrayList'e "add" işlemi yaptım.

## 6. Test Cases

Q1-> 10 bin integer içeren string "numbers.txt" ile denedim.

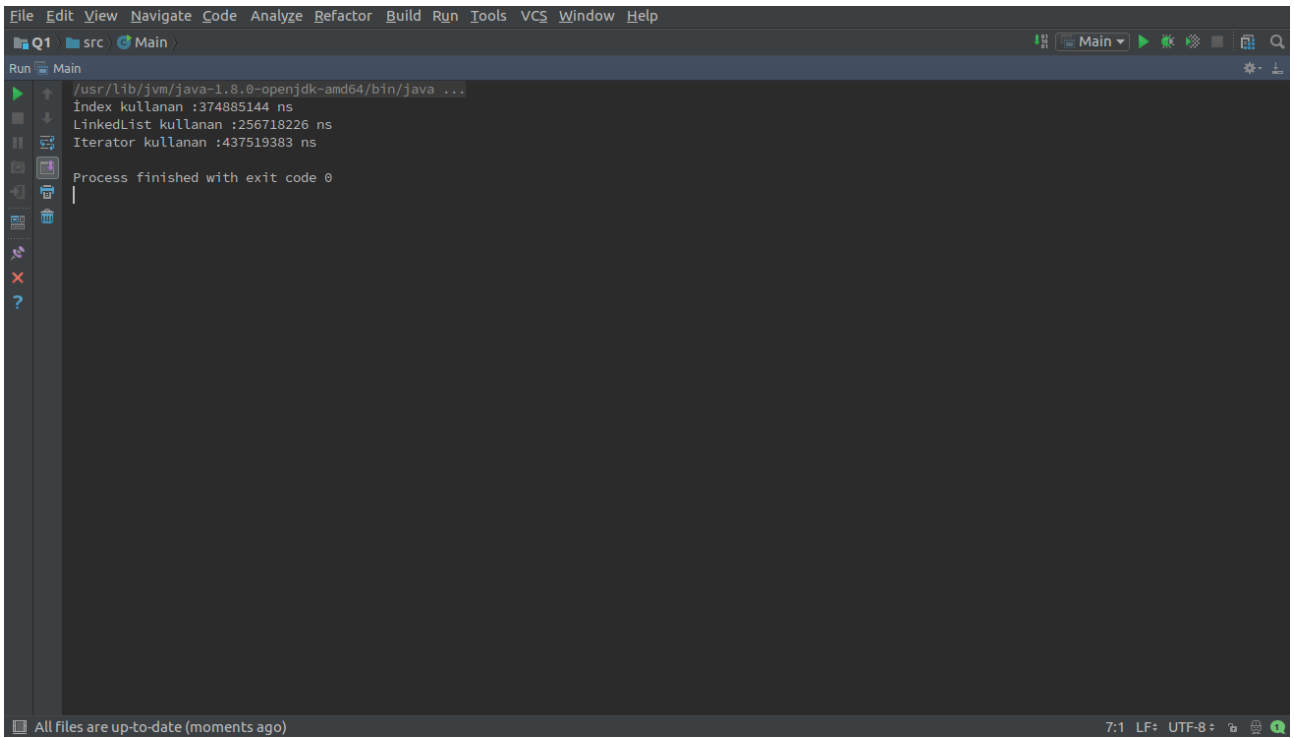
Q2-> Linkedlist'e belli sayıda "string" ekledim ve reverseToString metodunu çağırıp ekrana yazdırdım ve kontrol ettim.

Q3-> Test edilmeyecek.

Q4-> 100 integer içeren "numbers.txt" kullandım. Linkedlist'e 100 sayı ekledim. Ekrana yazdırdım. Sonrasında ilk 50 Node'u sildim. Ekrana yazdırdım. Sonrasında tekrar 100 integer ekledim. Ekrana yazdırdım. Kontrol ettim.

## 7. Running and Results

Q1) Input size = 10 000 iken performans sıralaması şu şekilde: Linkedlist < Index < Iterator



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Q1 src Main
Run Main
/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
Index kullanan :374885144 ns
LinkedList kullanan :256718226 ns
Iterator kullanan :437519383 ns
Process finished with exit code 0
All files are up-to-date (moments ago) 7:1 LF: UTF-8
```

RESULT : Yapılan performans ölçümlerine göre 10 000 integer içeren txt dosyasıyla çalıştırıldığında performans sıralaması şu şekilde oluyor:

-----LinkedList < Index < Iterator-----

Bu sıralamaya göre yorum yapacak olursak LinkedList diğer 2 implementasyona göre çok daha hızlı çalışmaktadır.Index ile yapılan implementasyon LinkedList'ten yavaş fakat Iterator implementasyonundan hızlıdır.En yavaş implementasyon ise Iterator'dur.

NOT: Yapılan ölçümler nanosaniye cinsinden olup 3 türde ki ToString() metodu karşılaştırılmıştır.