



**FATİH  
SULTAN  
MEHMET**  
VAKIF ÜNİVERSİTESİ

## **DATA STRUCTER HOMEWORK RAPOR**

<b>Raporu Hazırlayan:</b>	Merve Çınar
<b>Rapor Teslim Tarihi:</b>	14.05.2021

Soruda benden 15 tane öğrenciyi random bir not ile oluşturmam gereklidir ve oluştururken Single Link List özellikleri ile adding Removing özellikleri eklenmeli İstenilen List oluşturulmalı sayısı tutulmalı ve bu öğrencilere soruda verildiği gibi harf notu atamam gereklidir. Eğer print\_names e girilen harf notu ile öğrencinin aldığı harf notu eşit ise bunu outputa basması istenmiş.

**1)Student Sınıfım:** İlk olarak Student sınıfımı tanımladım bu sınıfım içerisinde soruda benden öğrenci adı ve puanı tutulması istendiği için değişkenlerimi tanımladım ve Her seferinde bir sonraki öğrenciyi gösterebilmek adına NextStudent tanımladım bu classımda constructor yazmak zorundayım çünkü CourseGrade sınıfından bu değerlere diğer türlü ulaşamam nextstudent da varsayılan olarak boş atadım.

## 2)CourseGrade Sınıfım:

**A)** Bu sınıfta öncelikle soruda benden istenilen değişkenleri tanımladım LinkedList listesi oluşturdum aynı arraylist gibi ardından ders adını tutması için coursname öğrenci numarası nı tutması için studentnumber tanımladım ve Benim Tek yönlü Linklistimin en baştaki elemanını tutması için Student classından head isimli bir değişken oluşturdum.

**B)**İkinci olarak bana bu class taki variable lerim için getter ve setter kodlarımı insert ettim.burada içinde değer yazma ve okuma işlevi yapılır

```
public LinkedList<Student> getStd() {
    return std;
}

public void setStd(LinkedList<Student> std) {
    this.std = std;
}

public void setCoursName(String coursName) {
    this.coursName = coursName;
}

public void setStudentsNumber(int studentsNumber) {
    this.studentsNumber = studentsNumber;
}

public String getCoursName() {
    return coursName;
}

public int getStudentsNumber() {
    return studentsNumber;
}
```

**C)** üçüncü olarak ise problemde ekleme metodlarının eklenme istenmiş bunun için Link Liste 3 adet ekleme fonksiyonu 1) addLast, 2)addFirst, 3) insertAfter.

1)addLast metodu sona öğrenci ekleme işine yarar. Burada head den başlayarak bir sonraki null elemanına kadar aranır böylece son eleman bulunur. Ve Kullanıcının girdiği isim son elemanın bir sonraki elemanı gösterir yani listenin sonuna eklenmiş olur

```
void addLast(String name) { //adding the data a
    Student newStudent = new Student(name); //
    if (head == null) { //if head=null the lis
        head = newStudent;
    } else { //this condition's main the list
        Student temp = head; //I have to do th
        while (temp.nextStudent != null) { //I
            temp = temp.nextStudent; //it fin
        }
        temp.nextStudent = newStudent; //end t
    }
    studentsNumber++; //I increase the number
    std.add(newStudent); // also I adding in s
}

public static void main(String[] args)
CourseGrades liste=new CourseGrades();
liste.addLast("MERVE");
liste.addLast("FATMA");
liste.addLast("NURDAN");
liste.addLast("EMİR");
liste.print();
}
```

```
Output - DataHomework (run) X
run:
MERVE -> FATMA -> NURDAN -> EMİR -> null
BUILD SUCCESSFUL (total time: 0 seconds)
```

2)addFirst methodu listenin başına girilen stringi ekler. En başta head olduğu için yeni eklenenin bir sonraki elemanını headi işaret ettirdikten sonra tekrar headi güncelleyip yeni eklediğim eleman yapıyorum.

```
void addFirst(String name) { // adding data beg
    Student newStudent = new Student(name); //
    if (head == null) { // if head=null the lis
        head = newStudent; //
    } else { //this mean, this list is not emp
        newStudent.nextStudent = head; // new
        head = newStudent; // and then I updat
    }
    studentsNumber++; //I increase the number
    std.add(newStudent); //also I adding in st
}

11 //
12 public class Test {
13     public static void main(String[] args) {
14         CourseGrades liste=new CourseGrades();
15         liste.addFirst("MERVE");
16         liste.addFirst("FATMA");
17         liste.addFirst("NURDAN");
18         liste.addFirst("EMİR");
19         liste.print();
20     }
21 }
```

```
Output - DataHomework (run) X
run:
EMİR -> NURDAN -> FATMA -> MERVE -> null
```

3) insertAfter insert kodunda iki parametre alıyor. İlk parametre eklenecek elemanı ikinci parametre hangi elemandan sonra ekleneceğini ifade eder.

```
void insertAfter(String name, String search) { //this f
    Student temp = head;
    while (temp != null && temp.name != search) {
        //this is searching the user entered string
        temp = temp.nextStudent; // this temp keep the
    }
    if (temp != null) { // I check it is not null
        Student newStudent = new Student(name); //creat
        newStudent.nextStudent = temp.nextStudent; //na
        temp.nextStudent = newStudent; //adding the stu
        std.add(newStudent); //ading atd LinkList
    } else {
        addLast(name); //if the search element null ,
    }
    studentsNumber++; //increases the number of student
}

public class Test {
    public static void main(String[] args) {
        CourseGrades liste=new CourseGrades();
        liste.addFirst("MERVE");
        liste.insertAfter("NURDAN", "MERVE");
        liste.insertAfter("YUSUF", "NURDAN");
        liste.insertAfter("EMİR", "NURDAN");
        liste.print();
    }
}

Output - DataHomework (run) X
run:
MERVE -> NURDAN -> EMİR -> YUSUF -> null
BUILD SUCCESSFUL (total time: 0 seconds)
```

D) Şimdide soruda istenildiği gibi silme metodlarını ekledim. Bir LinkedListten 3 çeşit silme olur.

1)Remove,2)RemoveLast,3)removeFirst.

1)Remove :Bir parametre alınır o parametreye girilen değer listede aranır ve bulunduğunda listeden silinir.

```
void remove(String name) { //It removes the data which user write as parameter.
    if (head == null) { //head=null this list no have data
        System.out.println("The list is empty, there is nothing to remove Student");
    } else if (head.name == name) {
        //if the first element equal user's enterd name and then I call the removeFi
        removeFirst();
    } else {
        Student temp = head; //I creat temp to don't lose head
        while (temp.nextStudent != null && temp.nextStudent.name != name) {
            //this while search the element which user entered.
            temp = temp.nextStudent;
        }
        if (temp.nextStudent != null && temp.nextStudent.name == name) {
            //I check the it is not null and it is equal the search data. if it is;
            std.remove(temp.nextStudent); //I remove std LinkList.
            temp.nextStudent = temp.nextStudent.nextStudent;
            //shows its value to the next element so that the search element disappe
            studentsNumber--; //reduce the number of students,because one student del
        }
    }
}

public class Test {
    public static void main(String[] args) {
        CourseGrades liste=new CourseGrades();
        liste.addFirst("MERVE");
        liste.insertAfter("NURDAN", "MERVE");
        liste.insertAfter("YUSUF", "NURDAN");
        liste.insertAfter("EMİR", "NURDAN");
        liste.remove("NURDAN");
        liste.print();
    }
}

Output - DataHomework (run) X
run:
MERVE -> EMİR -> YUSUF -> null
BUILD SUCCESSFUL (total time: 0 seconds)
```

2)removeLast: LinkedListin en sonundaki elemanı siler outputta baktığımızda en son Ethem eklendiği halde yok yani silindi.

```
void removeLast() { //Delete data from end of the list.
    if (head == null) { //if the head = null, the list is empty
        System.out.println("The list is empty, there is nothing to remove Student");
    } else if (head.nextStudent == null) { //this mean there is a one data on my list.
        System.out.println(head.name + " is removed in this list");
        // delete only one data which in the list.
        std.remove(head); //Delete from std LinkList
        studentsNumber--; //reduce the number of students.
        head = null; //this means now the list is empty
    } else { //the list is not empty and not there is a one data
        Student temp = head;
        while (temp.nextStudent.nextStudent != null) {
            //it finds the last element of List
            temp = temp.nextStudent;
        }
        System.out.println(temp.nextStudent.name + " is removed in this list ");
        std.remove(temp.nextStudent); //Delete from std LinkList
        studentsNumber--; //reduce the number of students.
        temp.nextStudent = null; //Last elements gone.
    }
}

public class Test {
    public static void main(String[] args) {
        CourseGrades liste=new CourseGrades();
        liste.addFirst("MERVE");
        liste.addLast("ETHEM");
        liste.insertAfter("NURDAN", "MERVE");
        liste.insertAfter("YUSUF", "NURDAN");
        liste.insertAfter("EMİR", "NURDAN");
        liste.removeLast();
        liste.print();
    }
}

Output - DataHomework (run) X
run:
ETHEM is removed in this list
MERVE -> NURDAN -> EMİR -> YUSUF -> null
BUILD SUCCESSFUL (total time: 0 seconds)
```

3)RemoveFirst:LinkedListin headini siler outputta görüldüğü gibi en başta Merve var ama silindi.

```
void removeFirst() { //delete the data from begin.
    if (head == null) { //if the list is empty it writes this message.
        System.out.println("The list is empty, there is nothing to remove Student");
    } else { //else the list is not empty, firstly it show which data delete and then
        System.out.println(head.name + " is removed in this list. "); //I write before
        std.remove(head); //Delete from std LinkList
        head = head.nextStudent; //now head show next element in this way first elemen
        studentsNumber--; //reduce the number of students.
    }
}

public class Test {
    public static void main(String[] args) {
        CourseGrades liste=new CourseGrades();
        liste.addFirst("MERVE");
        liste.addLast("ETHEM");
        liste.insertAfter("NURDAN", "MERVE");
        liste.insertAfter("YUSUF", "NURDAN");
        liste.insertAfter("EMİR", "NURDAN");
        liste.removeFirst();
        liste.print();
    }
}

Output - DataHomework (run) X
run:
MERVE is removed in this list.
NURDAN -> EMİR -> YUSUF -> ETHEM -> null
BUILD SUCCESSFUL (total time: 0 seconds)
```

E)randomGrade: tüm LinkedListin elemanlarına rastgele not atar ve harf notunu ve normal notunu basar.

```

void randomGrade() { // i made this function to see all student's grade and letter grade so i
char Letter = 'A'; // it is the default.
for (int i = 0; i < std.size(); i++) {
    //this is create the random grade between 0 and 100
    Random r = new Random();
    std.get(i).grade = r.nextInt(101);
    // The letter grade is assigned according to the random number formed by the following
    if (std.get(i).grade > 0 && std.get(i).grade < 60) {
        Letter = 'F';
    } else if (std.get(i).grade >= 60 && std.get(i).grade < 75) {
        Letter = 'C';
    } else if (std.get(i).grade > 75 && std.get(i).grade < 90) {
        Letter = 'B';
    } else if (std.get(i).grade >= 90 && std.get(i).grade < 100) {
        Letter = 'A';
    }
    // it is print the grades and letter grades
    System.out.println(std.get(i).name + " -> " + std.get(i).grade + " -> " + Letter);
}
}
}

```

```

----- Students ,their grades and their LetterGrade
NURDAN -> 69 -> C
MİRAC -> 64 -> C
SÜHEYLA -> 81 -> B
EMİR -> 0 -> B
KEVSER -> 89 -> B
MUSAB -> 35 -> F
ÖMER -> 91 -> A
ÇINAR -> 57 -> F
RÜVEYDA -> 38 -> F
MERYEM -> 51 -> F
EMRE -> 99 -> A
SEBAHAT -> 93 -> A
CİHANGİR -> 31 -> F
KEREM -> 7 -> F
MUSTAFA -> 92 -> A

```

**F)print\_name :** outputta random grade de notlar atanır ve tüm öğrenciler gösterilir. Print\_name bu notları baz alır ve içine girilen Char harf notuna eşit ise o notu alanı ve harf notunu basar. Kendim randomGrade den doğruluğunu kontrol ediyorum.

```

void print_name(char x) {
    //it works for randomGrade() function. it keep the which Grade char entered
    // and then it is search the LinkedList who take this Grade
    //then, it write the people who take entered
    char LetterGrade = 'A';

    for (int i = 0; i < std.size(); i++) {
        if (std.get(i).grade > 0 && std.get(i).grade < 60) {
            LetterGrade = 'F';
        } else if (std.get(i).grade >= 60 && std.get(i).grade < 75) {
            LetterGrade = 'C';
        } else if (std.get(i).grade > 75 && std.get(i).grade < 90) {
            LetterGrade = 'B';
        } else if (std.get(i).grade >= 90 && std.get(i).grade < 100) {
            LetterGrade = 'A';
        }
        if (LetterGrade == x) {
            // if the student's grade letter to user entered letter then it is print
            System.out.println(std.get(i).name + " take " + x + " grade.");
        }
    }
}

```

```

----- WHO Take B Letter Grade? -----
SÜHEYLA take B grade.
EMİR take B grade.
KEVSER take B grade.

```

**G)Hem linkListini normal hemde Array gibi olan LinkListi yazdırdım** ikisindeki elemanlar ve eleman sayısı aynı tabi ama ikisindeki sıralama da farklı çünkü Array e direk eklenir ken Link liste başa ekle çıkar gibi sıralama yapabiliriz.Ben iki çeşitte yazdırdım.

```

void print() {
    Student temp = head;
    while (temp != null) {
        System.out.print(temp.name + " -> ");
        temp = temp.nextStudent;
    }
    System.out.println("null");
}

void printL() {
    for (int i = 0; i < std.size(); i++) {
        System.out.println(std.get(i).name);
    }
}

```

```

public class Test {
    public static void main(String[] args) {
        CourseGrades list=new CourseGrades();
        list.addFirst("MERVE");
        list.addLast("ETHEM");
        list.remove("ETHEM");
        list.insertAfter("NURDAN", "MERVE");
        list.removeFirst();
        list.insertAfter("YUSUF", "NURDAN");
        list.removeLast();
        list.insertAfter("EMİR", "NURDAN");
        list.addFirst("KEVSER");
        list.addLast("MİRAC");
        System.out.println("-----");
        list.print();
        System.out.println("-----");
        list.printL();
    }
}

```

```

Output - DataHomework (run) X
-----
ETHEM is removed in this list
MERVE is removed in this list
YUSUF is removed in this list
-----
KEVSER -> NURDAN -> EMİR -> MİRAC -> null
-----
NURDAN
EMİR
KEVSER
MİRAC

```

**3)Test Sınıfı:** Son olarak ödevimizde 15 kişilik bir sınıf oluşturmamızı ve bu öğrencilere random değer atayıp harf notu ile derecelendirilmemiz istenmiş ve print\_name ile içeri yazılan harfi not olarak alan öğrencinin yazılması istenmişti bu yüzden tüm metodlarımızı bastıran bir output hazırladım randomGrade ile random değer atıp 15 öğrencinin notlarını görebiliriz ve print\_name ile B notu alan öğrencileri sarı fosforlu bölgede görebilirsiniz. Hem Linklist olarak hem Std Link List olarak öğrenciler listelenmiştir. Hem size metodu ile hem std List deki student number ile de öğrenci sayısı bastırılmıştır.

```

CLASS: DATA STRUCTER
----- Print LinkList -----
MUSTAFA -> CİHANGİR -> SEBAHAT -> ÇINAR -> MUSAB -> KEVSER -> SÜHEYLA -> NURDAN -> EMİR -> MİRAC -> ÖMER -> MERYEM -> RÜVEYDA -> EMRE -> KEREM -> nul:
-----Print std LinkList-----
NURDAN
MİRAC
SÜHEYLA
EMİR
KEVSER
MUSAB
ÖMER
ÇINAR
RÜVEYDA
MERYEM
EMRE
SEBAHAT
CİHANGİR
KEREM
MUSTAFA
----- Students ,their grades and their LetterGrade -----
NURDAN -> 79 -> B
MİRAC -> 27 -> F
SÜHEYLA -> 55 -> F
EMİR -> 38 -> F
KEVSER -> 86 -> B
MUSAB -> 9 -> F
ÖMER -> 70 -> C
ÇINAR -> 56 -> F
RÜVEYDA -> 37 -> F
MERYEM -> 10 -> F
EMRE -> 98 -> A
SEBAHAT -> 86 -> B
CİHANGİR -> 77 -> B
KEREM -> 8 -> F
MUSTAFA -> 89 -> B
----- WHO Take B Letter Grade? -----
NURDAN take B grade.
KEVSER take B grade.
SEBAHAT take B grade.
CİHANGİR take B grade.
MUSTAFA take B grade.
-----
Student Numbers std LinkList = 15 , Student Numbers normal LinkList = 15

```