

Rapor

Problem; Öğrenci adlarını (veya kimlik numaralarını) ve notlarını ($0 \leq \text{derece} \leq 100$) tutan bir Öğrenci sınıfı tanımlayın. Ayrıca, belirli bir kursun öğrenci listesini izleyen bir CourseGrades sınıfı tanımlayın. Sınıfta ders adı, öğrenci sayısı ve öğrenci listesi için alanlar ve bu alanlara erişim ve yazdırma için erişimci yöntemler bulunmalıdır. Sınıfta ayrıca listeye öğrenci ekleme / listeden öğrenci çıkarma yöntemleri ve harf notları L'ye eşit olan öğrenci adlarını bulup yazdıran bir print_names (L) yöntemi olacaktır. Tek Bağlantılı Liste kullanarak öğrenci listesini uygulayın.

Her iki uygulamanın da doğru çıktılar ürettiğini göstermek için TestGrades adlı bir test kodu yazın:
Test Senaryosu: Rastgele oluşturulmuş notlara sahip 15 öğrenciden oluşan kurs:

- A, B, C ve F harf notları için print_names yöntemini çağırın.

İlk olarak Studen sınıfını oluşturdum ve buraya name, grade ve next özelliklerini tanımladım. Sonrasında CourseGrade sınıfında course_name , numberOfStudent ve head özelliklerini tanımladım. Student ve Course sınıfında tanımladığım next ve head özellikleri Linked List özelliği kullanabilmek için, ve son olarak courseGrade sınıfında get, set ve print metodlarımı tanımladım.

```
public class CourseGrades {  
  
    String course_name; //kursun ismini ve  
    int numberOfStudent; // kurstaki öğrenci sayısını hesapla  
    LinkedList<Student> students=new LinkedList<>(); //öğrenci listesi  
    Student head; //listeyi link list olarak tanımladığım  
  
    public String getCourse_name() {  
        return course_name;  
    }  
  
    public void setCourse_name(String course_name) { //kursun ismini ve  
        this.course_name = course_name;  
    }  
  
    public int getNumberOfstudent() { //öğrenci sayısını bul  
        return numberOfStudent;  
    }  
  
    public void setNumberOfstudent(int numberOfstudent) {  
        this.numberOfStudent = numberOfstudent;  
    }  
}  
  
public class TestGrades {  
    public static void main(String[] args) {  
        CourseGrades liste=new CourseGrades();  
  
        liste.setCourse_name("Math course");  
        liste.printCoursName();  
        System.out.println("size -> "+liste.getNumberOfstudent());  
    }  
}
```

```
* @author MERVE  
*/  
  
public class Student {  
    String name;  
    Student next;  
    int grade;  
  
    public Student(String name) {  
        this.name=name;  
        this.grade=grade;  
        this.next = null;  
    }  
}
```

```
Output - data_hmw1 (run)  
run:  
Course name=Math course  
size -> 0  
BUILD SUCCESSFUL (total time: 0 seconds)
```

Hem Link listemi hemde CourseGrade sınıfında ayrıca tanımladığım studens listemi yazdırmak için print fonksiyonu

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

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

}
```

Listenin en başına eklemek için addFirst fonksiyonu

```
void addFirst(String name){
    Student newStudent=new Student(name); //1
    if(head==null){ //eğer listenin başı boş :
        head=newStudent;
    }
    else{
        newStudent.next=head; //liste boş değ
        head=newStudent; //baş kısmını da yen
    }

    students.addFirst(newStudent); //yukarıda
    numberOfstudent++; //öğrenci sayısını b
```

```
liste.addFirst("asude");
liste.addFirst("ayse");
liste.addFirst("ahmet");
liste.print();
liste.printList();
```

```
Output - Java IntelliJ (run)
run:
ahmet -> ayse -> asude -> null
ahmet
ayse
asude
```

Listenin en sonuna ekleme addLast fonksiyonu

```
void addLast(String name){
    Student newStudent=new Student(name); //1

    if(head==null){ //liste boş ise tekrar ye
        head=newStudent;
    }
    else{ //liste boş değil ise
        Student temp=head; //geçici bir değ
        while(temp.next!=null){ // listenin s
            temp=temp.next;
        }
        temp.next=newStudent; //temp'in next'
        // yeni öğrenci
    }

    students.addLast(newStudent); //yukarıda
    numberOfstudent++; //öğrenci sayısını bi
}

liste.addFirst("asude");
liste.addFirst("ayse");
liste.addFirst("ahmet");
liste.addLast("ömer");
liste.addLast("eylül");
liste.print();
liste.printList();
```

```
run:
ahmet -> ayse -> asude -> ömer -> eylül -> null
ahmet
ayse
asude
ömer
eylül
```

Listenin ortasına ekleme için insertAfter fonksiyonu

```
void insertAfter(String name,String search){
    Student temp=head;
    int indeks;

    while(temp!=null && temp.name!=search){ //temp
        temp=temp.next; //liste
    }

    if(temp!=null){
        Student newStudent=new Student(name); //li
        newStudent.next=temp.next; //ge
        temp.next=newStudent; //ge
        indeks= students.indexOf(newStudent.next); //ye
        students.add(indeks, newStudent); //ve inde

        liste.addFirst("ahmet");
        liste.addLast("ömer");
        liste.addLast("eylül");
        liste.insertAfter("emre", "ahmet");
        liste.print();
        liste.printList();

        ahmet -> emre -> ömer -> eylül -> null
        ahmet
        emre
        ömer
        eylül
    } else{
        addLast(name); // diğer koşulda ise direkt
        liste.print();
        liste.printList();

        emre
        ömer
        eylül
    }

    numberOfstudent++; //öğrenci sayısını bir arttı
}
```

Listenin en sonundan silmek için removeLast fonksiyonu

```
removeLast(){
    if(head==null)
        System.out.println("Listede silinecek eleman yok")
    else if(head.next==null){
        System.out.println(head.name+" silindi"); //eğer
        head=null;
    }
    else{
        Student temp=head;
        while(temp.next.next!=null){ //temp'in next'inin
            temp=temp.next; //tüm listeyi gezeri
        }
        System.out.println(temp.next.name + " silindi"); //
        temp.next=null; // temp'in nextinin null'a eşitle
        //ve listenin sonundaki kişi artı
        students.removeLast(); //students listemizin sonu
        liste.addFirst("ahmet");
        liste.addLast("ömer");
        liste.addLast("eylül");
        liste.print();
        liste.printList();
        liste.removeLast();
        liste.print();
        liste.printList();

        ahmet -> ömer -> eylül -> null
        ahmet
        ömer
        eylül
        eylül silindi
        ahmet -> ömer -> null
        ahmet
    }

    numberOfstudent--; //öğrenci sayısını bir azaltırız
}
```

Listenin başında silmek için removeFirst fonksiyonu

```
void removeFirst(){
    if(head==null) //liste boş ise işlem yapılmayacak
        System.out.println("Listede Silinecek eleman yok");
    else{
        System.out.println(head.name+" silindi"); //listeni
        head=head.next; //head'i head'in next'ine atarız
        students.removeFirst(); //students listemizden de
        liste.addFirst("ahmet");
        liste.addLast("ömer");
        liste.addLast("eylül");
        liste.print();
        liste.printList();
        liste.removeFirst();
        liste.print();
        liste.printList();

        run:
        ahmet -> ömer -> eylül -> null
        ahmet
        ömer
        eylül
        ahmet silindi
        ömer -> eylül -> null
        ömer
        eylül
        SUCCESSFUL (total time: 0
    }

    numberOfstudent--; //öğrenci sayısını bir azalttım
}
```

Listede direkt istenilen kişiyi silme (herhangi bir yerinden) remove fonksiyonu

```
d remove(String name){
if(head==null) //liste boş ise silme işlemi yapılmıyor
    System.out.println("there is no one to remove from the list");

else{
    Student temp=head; //listeyi gezmek için geçicici değişkn oluşturdum
    while(temp.next!=null && temp.next.name!=name){ //listenin null olana kadar ve :
        temp=temp.next; //listede ilerliyoruz
    }
    if(temp.next!=null && temp.next.name==name){ //eğer aradığımız isim geçici değ.
        System.out.println(temp.next.name+" silindi"); //silinecek olan kişinin ism.
        students.remove(temp.next); //students listemizden de kişiyi burada sildird.
        temp.next=temp.next.next; //link listede de geçici değişkenimizin sıradaki :
        //kişiyi listeden çıkarmış olduk
        numberOfStudent--; //öğrenci sayısını bir azaltırız

        liste.addFirst("asude");
        liste.addLast("merve");
        liste.addLast("eylül");
        liste.print();
        liste.printList();
        liste.remove("ayse");
        liste.remove("merve");
        liste.print();
        liste.printList();

        asude -> merve -> eylül -> null
        asude
        merve
        eylül
        ayse -> this person is not in the list, please check
        merve silindi
        asude -> eylül -> null
        asude
        eylül
    }
    else { //eğer aradığımız isim listede yok ise bir silme işlemi gerçekleşmeyecek
        //olmayacak

        System.out.println(name + " -> this person is not in the list, please check");
    }
}
```

Öğrencilere random not atama ve yazdırma randomGrades fonksiyonu

```
void randomGrades(){

    for (int i = 0; i < students.size(); i++) { //students listemizi for ile döndüm

        Random rnd=new Random(); //random sayı atamak için random fonksiyonunu kullandım
        students.get(i).grade=rnd.nextInt(100); //0-100 arası random sayıları students listesi
        if(students.get(i).grade>=0 && students.get(i).grade<60){ //eğer sayı 0 ve 60 arasında

            System.out.println(students.get(i).name+ " -> " + students.get(i).grade+ " -> F");
        }
        else if(students.get(i).grade>=60 && students.get(i).grade<75){ //eğer sayı 60 ve 70
            System.out.println(students.get(i).name+ " -> " + students.get(i).grade+ " -> C");
        }
        else if(students.get(i).grade>=75 && students.get(i).grade<90){ //eğer sayı 60 ve 70
            System.out.println(students.get(i).name+ " -> " + students.get(i).grade+ " -> B");//öğ
            //ha
        }
        else if(students.get(i).grade>=90 && students.get(i).grade<100){ //eğer sayı 60 ve 70
            System.out.println(students.get(i).name+ " -> " + students.get(i).grade+ " -> A"); //öğ
            //tekr

            liste.addFirst("asude");
            liste.addLast("merve");
            liste.addLast("eylül");
            liste.addFirst("emel");
            liste.addFirst("arda");
            liste.randomGrades();

            arda -> 60 -> C
            emel -> 28 -> F
            asude -> 91-> A
            merve -> 65 -> C
            eylül -> 56 -> F
            BUT ITD SUCCESSFULL. /
    }
}
```

Öğrencilerin harf notuna göre isim yazdıran print_names fonksiyonu

```
void print_names(char harfN){

    for (int i = 0; i < students.size(); i++){ //students listesindeki öğrencileri for döngüsü
    if(harfN=='A' &&students.get(i).grade >=90 ){ //eğer parametre olarak girilen harf notu A'ya
        System.out.println("who get "+ harfN+ " from course -> " + students.get(i).name); //öğren

        liste.addFirst("asude");
        liste.addLast("merve");
        liste.addLast("eylül");
        liste.addFirst("emel");
        liste.addFirst("arda");
        liste.randomGrades();

        arda -> 12 -> F
        emel -> 78 -> B
        asude -> 70 -> C
        merve -> 99-> A
        eylül -> 41 -> F
        -----A-----
        who get A from course -> merve
        -----B-----
        who get B from course -> emel
        -----C-----
        who get C from course -> asude
        -----F-----
        who get F from course -> arda
        who get F from course -> eylül
        G -> invalid letter

    }
    else if(harfN=='B' && students.get(i).grade >=75 && students.get(i).grade <90){ //eğer
        System.out.println("who get "+ harfN+ " from course -> " + students.get(i).name); //ve

        liste.addFirst("ardade");
        liste.addLast("merve");
        liste.addLast("eylül");
        liste.addFirst("emel");
        liste.addFirst("arda");
        liste.randomGrades();

        arda -> 12 -> F
        emel -> 78 -> B
        asude -> 70 -> C
        merve -> 99-> A
        eylül -> 41 -> F
        -----A-----
        who get A from course -> merve
        -----B-----
        who get B from course -> emel
        -----C-----
        who get C from course -> asude
        -----F-----
        who get F from course -> arda
        who get F from course -> eylül
        G -> invalid letter

    }
    else if(harfN=='C' &&students.get(i).grade >=60 && students.get(i).grade <75 ){//aynı
        System.out.println("who get "+ harfN+ " from course -> " + students.get(i).name); //tekr

        liste.addFirst("ardade");
        liste.addLast("merve");
        liste.addLast("eylül");
        liste.addFirst("emel");
        liste.addFirst("arda");
        liste.randomGrades();

        arda -> 12 -> F
        emel -> 78 -> B
        asude -> 70 -> C
        merve -> 99-> A
        eylül -> 41 -> F
        -----A-----
        who get A from course -> merve
        -----B-----
        who get B from course -> emel
        -----C-----
        who get C from course -> asude
        -----F-----
        who get F from course -> arda
        who get F from course -> eylül
        G -> invalid letter

    }
    else if(harfN=='F' && students.get(i).grade >=0 && students.get(i).grade <60 ){
        System.out.println("who get "+ harfN+ " from course -> " + students.get(i).name);

        liste.addFirst("ardade");
        liste.addLast("merve");
        liste.addLast("eylül");
        liste.addFirst("emel");
        liste.addFirst("arda");
        liste.randomGrades();

        arda -> 12 -> F
        emel -> 78 -> B
        asude -> 70 -> C
        merve -> 99-> A
        eylül -> 41 -> F
        -----A-----
        who get A from course -> merve
        -----B-----
        who get B from course -> emel
        -----C-----
        who get C from course -> asude
        -----F-----
        who get F from course -> arda
        who get F from course -> eylül
        G -> invalid letter

    }
    if(harfN!='A' && harfN!='B' && harfN!='C' && harfN!='F'){ //eğer geçersiz bir harf girilir
        System.out.println( harfN + " -> invalid letter");
    }
}
```