

## VERİ YAPILARI ARA SINAV

**Soru 1(30p)** Aşağıdaki yapı, 2. Ödevinizde kullanmanız gereken yapıdır. Soruları bu yapıya göre cevaplayınız. Yazacağınız metodlar bu yapının oluşturulduğu sınıf içerisinde olacaktır. Bu yapıda iki farklı düğüm(Node) sınıfı bulunmaktadır. NodeStudent sınıfı öğrenci bilgilerini tutmakta, NodeCourse sınıfı ders bilgilerini tutmaktadır. Sınıf yapıları şu şekildedir(sadece field'lar gösterilmiştir). (Answer the question based on the node structure shown below. There are two different node structures. Class fields are shown below.)

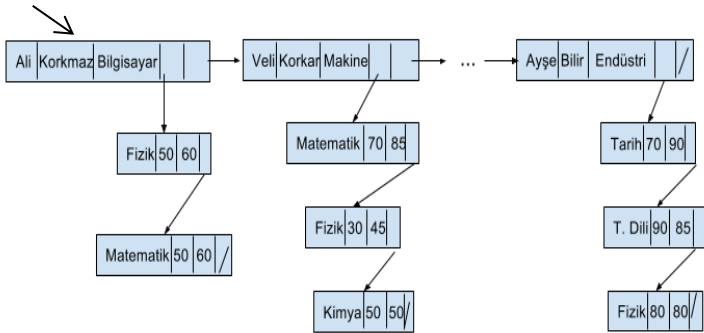
*Class NodeStudent{ public Student value, public NodeStudent next, public NodeCourse nextCourse ...}*

*Class NodeCourse{public Course value, public NodeCourse next ...}*

*Class Student{ public String name, public string surname public string department ...}*

*Class Course{public string name, public int midterm, public int final...}*

head



- a) Verilen bir bölümde okuyan öğrencilerin isimlerini ve soyisimlerini listeleyen metodu yazınız. (display the names and last names of the students of a given department)(10p)

```
Public void DisplayStudents(string bolum){
```

```
}
```

- b) En fazla derse kayıt olan öğrencinin ismini bulan metodu yazınız.(find the name of the student who registered to the maximum number of courses) (20p)

```
Public string FindStudent(){
```

}

**Soru 2(20p)** Dairesel(circular) bir çift bağılı listeyi, normal çift bağılı listeye dönüştüren metodu yazınız. Bu metod generic DoublyLinkedList sınıfının içinde olacak. DNode<T> tipinde olan head ilk elemanı göstermektedir. (Write a method that converts circular doubly linked list into regular doubly linked list. Head shows the first node in the list and its type is DNode<T> )  
*Class DNode<T>{ T value, DNode<T> next, DNode<T> prev ...}*

**Soru 3(20p)** Aşağıdaki değerleri AVL ağacına ekleyiniz, adımları gösteriniz ve inorder, preorder ve postorder yöntemleriyle elemanları listeleyiniz. Daha sonra 40'ı siliniz. (Insert the following values into AVL tree and traverse it with inorder, preorder, postorder. Then delete 40 from the tree.)

35, 15, 8, 20, 55, 40, 18, 16, 22

**Soru 4(20p)** Generic bir ikili arama ağacı üzerinde verilen bir değer için inorder predecessor(inorder dolaşma yöntemine göre kendisinden önce gelen değer) değerini bulunuz. Aşağıdaki metodlar sizin kullanmanız için hazırdır.(find an inorder predecessor of a given value on a binary search tree. Inorder predecessor is the one that comes just before the given value in an inorder traversal. You can use the following methods in your code).

**Public TNode<T> findNode(T val); //verilen değer bulduğu düğümü döndürür.**

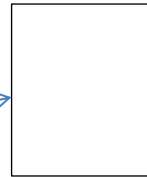
**Public TNode<T> findParent(T val); //Verilen değer ebeveyninin bulunduğu düğümü döndürür.**

**Public T FindInOrderPredecessor(T val){**

**}**

**Soru 5(20p)** Aşağıdaki kod için fonksiyonların ne işe yaradığını belirten commentleri ekleyin ve programın çıktısını yandaki kutucuğa yazınız. (comment on what the following methods do. What is the output of the following program.)

```
public class Node{
    public int deger;
    public Node next;
    public Node(int deger)
    {
        this.deger = deger;
    }
}
class LinkedList
{
    public Node head;
    public void fonksiyon1()
    {
        Node iterator = head;
        Node prev;
        if (head==null || head.next == null)
        {
            head = null;
        }
        else
        {
            prev = null;
            while (iterator.next != null)
            {
                prev = iterator;
                iterator = iterator.next;
            }
            prev.next=new Node(prev.deger);;
        }
    }
    public void fonksiyon2(int x)
    {
        Node iterator = head;
        while (iterator.next != null)
            iterator = iterator.next;
        (new Node(x)).next = head;
        iterator.next = new Node(x);
    }
    public void fonksiyon3(int x)
    {
        if (head == null)
            head = new Node(x);
        else
        {
            Node temp = new Node(x);
            temp.next = head.next;
            head.next = temp;
        }
    }
    public void fonksiyon4(){
        Node iterator = head.next;
        while (iterator != null)
        {
            Console.WriteLine(iterator.deger);
            iterator = iterator.next;
        }
    }
    static void Main(string[] args)
    {
        LinkedList db = new LinkedList();
        db.fonksiyon3(1);
        db.fonksiyon3(3);
        db.fonksiyon2(2);
        db.fonksiyon3(4);
        db.fonksiyon1();
        db.fonksiyon2(5);
        db.fonksiyon4();
    }
}
```



Time: 105 minutes

Başarılar

Yrd. Doç. Dr. Gürhan Gündüz