**FİNALE DÖNÜK ÇALIŞMALAR:**

**SORU1 : AĞAÇ VERİ YAPISI**

**package trees;**

**/\*İkili Arama Ağacı Oluşturmayı ve**

**Dolaşmayı Sağlayan Java Programı**

**\*/**

**public class Dugum {//Dugum**

**public int data;**

**public Dugum leftChild;**

**public Dugum rightChild;**

**public void displayNode()**

**{**

**System.out.print(" "+data+" ");**

**}**

**}**

**package trees;**

**public class Agac { //Ağaç**

**private Dugum root;**

**public Agac() //Yapıcı Metot**

**{**

**root = null;**

**}**

**public Dugum getRoot()**

**{**

**return root;**

**}**

**// Aağcın preOrder Dolaşılması**

**public void preOrder(Dugum localRoot)**

**{**

**if(localRoot!=null)**

**{**

**localRoot.displayNode();**

**preOrder(localRoot.leftChild);**

**preOrder(localRoot.rightChild);**

**}**

**}**

**// Ağacın postOrder Dolaşılması**

**public void postOrder(Dugum localRoot)**

**{**

**if(localRoot!=null)**

**{**

**postOrder(localRoot.leftChild);**

**postOrder(localRoot.rightChild);**

**localRoot.displayNode();**

**}**

**}**

**// Aağca bir düğüm eklemeyi salayan metot**

**public void insert(int newdata)**

**{**

**Dugum newNode = new Dugum();**

**newNode.data = newdata;**

**if(root==null)**

**root = newNode;**

**else**

**{**

**Dugum current = root;**

**Dugum parent;**

**while(true)**

**{**

**parent = current;**

**if(newdata<current.data)**

**{**

**current = current.leftChild;**

**if(current==null)**

**{**

**parent.leftChild=newNode;**

**return;**

**}**

**} else**

**{**

**current = current.rightChild;**

**if(current==null)**

**{**

**parent.rightChild=newNode;**

**return;**

**}**

**}**

**} // end while**

**} // end else not root**

**} // end insert()**

**} // class Tree**

**package trees;**

**public class TestAgac {**

**public static void main(String [] args)**

**{**

**Agac agac = new Agac();**

**// Aağca 10 tane sayı yerleştirilmesi**

**System.out.println("Sayılar : ");**

**for (int i=0;i<10;++i) {**

**int sayi = (int) (Math.random()\*100);**

**System.out.print(sayi+" ");**

**agac.insert(sayi);**

**};**

**System.out.print("\nAğacın PreOrder Dolaşılması : ");**

**agac.preOrder(agac.getRoot());**

**System.out.print("\nAğacın PostOrder Dolaşılması : ");**

**agac.postOrder(agac.getRoot());**

**}**

**}**

**SORU2: DİNAMİK DİZİLER**

**Dinamik Diziler başta belirlediğimiz boyut için bir çözüm sunmaktadır. İş dinamiklik kazanacağından beraberinde esnekliğinde (flexibility) sağlayacaktır. Bu durumda hala daha en basit veri yapısı olan dizileri daha bir esnek şekilde kullanabiliyoruz.**

5

5

**Yeni eleman ekleme**

**array Copy new element**

**Yeni eleman ekleme**

20

5

20

**array=temp**

-2

20

5

**Yeni eleman ekleme**

-2

**Dinamik dizileri kullanma amacımız: bellek yönetiminin iyi idare edilmesini sağlamak. Dizi boyutunun program çalışması sırasında artıp azalmasını sağlamak.**

**package dynamicarrays;**

**import java.util.Arrays;**

**public class DynamicArrays {**

**private int array[];**

**private int size; //Dizinin mevcut boyutunu tutar**

**private int capacity;//Dizinin toplam kapasitesini tutar**

**//Dizi ve değerleri başlatmak için varsayılan yapıcı (Constructor)**

**public DynamicArrays(){**

**array=new int[2];**

**size=0;**

**capacity=2;**

**}**

**public void ensureCapacity(int mincapacity)**

**{**

**int temp[]=new int[capacity\*mincapacity];**

**for(int i=0;i<capacity;i++)**

**{**

**temp[i]=array[i];**

**}**

**array=temp;**

**capacity=capacity\*mincapacity;**

**}**

**//Sonuna eleman eklemek için**

**public void addElement(int element)**

**{**

**if(size==capacity)//Eğer tahsis edilen alanın tamamı kullanıldıysa**

**{**

**ensureCapacity(2);//Kapasiteni 2 katına çıkar**

**}**

**array[size]=element;**

**size++;**

**}**

**//Bir indeksteki elemanı elde etmek için**

**public int getElement(int index)**

**{**

**return array[index];**

**}**

**//Bir indeksteki elementi silmek**

**public void removeElement(int index)**

**{**

**if(index>=size || index<0)**

**{**

**System.out.println("İlgili indexte herhangi bir eleman bulunmamaktadır");**

**}**

**else**

**{**

**for(int i=index;i<size-1;i++)**

**{**

**array[i]=array[i+1];**

**}**

**array[size-1]=0;**

**size--;**

**}**

**}**

**public int size()**

**{**

**return size;**

**}**

**public int Capacity()**

**{**

**return capacity;**

**}**

**//Dizideki elemanları ekranda göstermek**

**public void PrintElements()**

**{**

**System.out.println("Dizinin Elemanları: "+Arrays.toString(array));**

**}**

**}**

**package dynamicarrays;**

**public class DynamicArrayTest {**

**public static void main(String[] args)**

**{**

**DynamicArrays d\_array=new DynamicArrays();**

**//0. ve 1. indekslerine eleman ekleme**

**d\_array.addElement(63);**

**d\_array.addElement(42);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**d\_array.addElement(21);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**d\_array.addElement(86);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**d\_array.addElement(53);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**d\_array.removeElement(2);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**d\_array.removeElement(1);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**d\_array.removeElement(2);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**d\_array.addElement(75);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**d\_array.removeElement(1);**

**System.out.println("Size:"+d\_array.size()+"and Capacity:"+**

**+d\_array.Capacity());**

**d\_array.PrintElements();**

**}**

**}**

**Size:2and Capacity:2**

**Dizinin Elemanları: [63, 42]**

**Size:3and Capacity:4**

**Dizinin Elemanları: [63, 42, 21, 0]**

**Size:4and Capacity:4**

**Dizinin Elemanları: [63, 42, 21, 86]**

**Size:5and Capacity:8**

**Dizinin Elemanları: [63, 42, 21, 86, 53, 0, 0, 0]**

**Size:4and Capacity:8**

**Dizinin Elemanları: [63, 42, 86, 53, 0, 0, 0, 0]**

**Size:3and Capacity:8**

**Dizinin Elemanları: [63, 86, 53, 0, 0, 0, 0, 0]**

**Size:2and Capacity:8**

**Dizinin Elemanları: [63, 86, 0, 0, 0, 0, 0, 0]**

**Size:3and Capacity:8**

**Dizinin Elemanları: [63, 86, 75, 0, 0, 0, 0, 0]**

**Size:2and Capacity:8**

**Dizinin Elemanları: [63, 75, 0, 0, 0, 0, 0, 0]**

**SORU3: ÇİFT YÖNLÜ BAĞLI LİSTE**

.previous V .next

.previous V .next

**Head Node**

**package doublelinkedlist;**

**public class Node {**

**Object value;**

**Node next;**

**Node previous;**

**public Node (Object n\_value, Node n\_next,Node n\_previous)**

**{**

**this.value=n\_value;**

**this.next=n\_next;**

**this.previous=n\_previous;**

**}**

**}**

**package doublelinkedlist;**

**public class DoubleLinkedList<T> {**

**Node headNode;**

**public DoubleLinkedList()//Constructor method varsayılan yapıcı metot**

**{**

**headNode=null;**

**}**

**public void add(Object val)**

**{**

**Node newnode=new Node(val, null, null);**

**if(headNode==null)**

**{**

**headNode=newnode;**

**}**

**else**

**{**

**newnode.next=headNode;**

**headNode.previous=newnode;**

**headNode=newnode;**

**}**

**}**

**public void delete()**

**{**

**headNode=headNode.next;**

**headNode.previous=null;**

**}**

**public void display()**

**{**

**Node n=headNode;**

**while(n!=null)**

**{**

**System.out.println(n.value);**

**n=n.next;**

**}**

**}**

**}**

**package doublelinkedlist;**

**public class DoubleLınkedListTest {**

**public static void main(String [] args)**

**{**

**DoubleLinkedList <String> d\_list=new DoubleLinkedList <>();**

**System.out.println("..................");**

**d\_list.add("Canan");**

**d\_list.add("BATUR");**

**d\_list.add("ŞAHİN");**

**d\_list.display();**

**System.out.println("..................");**

**d\_list.delete();**

**d\_list.display();**

**System.out.println("..................");**

**d\_list.delete();**

**d\_list.display();**

**System.out.println("..................");**

**}**

**}**