**VY-Ö3**

**1)Öğrencilerin isimlerini ve notlarını tutan tek yönlü bir bağlı liste oluşturunuz.**

public class Node {

String isim;

int not;

Node ileri;

public Node (String isim,int not) {

this.isim=isim;

this.not=not;

this.ileri=null; } }

public class TYbL {

Node bas=null,son=null;

public void basaekle(Node yeni) {

if (bas==null) {

bas=yeni;

son=yeni; }

else {

Node temp=bas;

while (temp.ileri!=null); {

temp=temp.ileri;} } }

public void yazdir () {

if (bas==null) {

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

return; }

Node temp=bas;

Node enbasarili=bas;

while (temp!=null) {

if (temp.not > enbasarili.not);{

enbasarili=temp;

temp=temp.ileri;}

System.***out***.println("En başarılı öğrenci= "+ enbasarili.isim + " "+ "\nNot= " + enbasarili.not); }}

public static void main (String[]args) {

TYbL I= new TYbL ();

Node s= new Node ("Deniz",65);

Node s1=new Node ("Aras",89);

Node s2=new Node ("Ali",87);

I.basaekle(s);

I.basaekle(s1);

I.basaekle(s2);

I.yazdir();

} }

**2) İndeks hesaplama.**

public class Node {

String isim;

double kilo;

double boy;

Node ileri;

public Node (String isim,double kilo, double boy) {

this.isim=isim;

this.kilo=kilo;

this.boy=boy;

this.ileri=null; }

public double Hesapla() {return kilo/ (boy\*boy); } }

public class TbL {

Node bas=null, son=null;

public void basaEkle(Node yeni) {

if (bas == null) {

bas =yeni; }

else { Node temp = bas;

while (temp.ileri!= null) {

temp = temp.ileri; }

temp.ileri = yeni; } }

public void yazdir() {

if (bas== null) {

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

return; }

Node temp = bas;

while (temp != null) {

double Hesapla = temp.Hesapla();

if (Hesapla >= 30) {

System.***out***.println(temp.isim + " Vücut Kitle İndeksi= " + Hesapla); }

temp = temp.ileri; }}

public static void main(String[] args) {

TbL I = new TbL();

Node s=new Node ("Ayşe", 45.0, 1.70);

Node s1=new Node("Ali", 78.0, 1.90);

Node s2=new Node("Ahmet", 56.0, 1.50);

Node s3=new Node("Veli", 65.0, 1.69);

Node s4=new Node("Kemal", 59.0, 1.63);

I.basaEkle(s);

I.basaEkle(s1);

I.basaEkle(s2);

I.basaEkle(s3);

I.basaEkle(s4);

I.yazdir(); }}

**3. 83’ü ekleme sorusu.**

package çözüm;

public class Node {

int sayı;

Node ileri;

Node geri;

int j;

Node (int sayı) {

this.sayı=sayı;

this.ileri=null;

this.geri=null;

this.j=j; } }

public class CbL {

Node bas=null;

public void siraliEkle(Node yeni) {

if (bas == null) {

bas = yeni;

return; }

Node m = bas;

while (m!= null && m.sayı <i) {

m =m.ileri; }

if (m == bas) {

yeni.ileri = bas;

bas.geri = yeni;

bas = yeni; }

else if (m == null) {

Node son = bas;

while (son.ileri != null) {

son = son.ileri;}

son.ileri = yeni;

yeni.geri = son }

else { Node ileri = m.geri;

m.ileri = yeni;

yeni.geri =m.ileri;

yeni.ileri = m;

m.geri = yeni ;} }

public void yazdir() {

Node m = bas;

while (m != null) {

System.***out***.print(m.sayı + " ");

m = m.ileri; }

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

public static void main(String[] args) {

CbL l = new CbL();

Node s=new Node (69);

Node s1=new Node(80);

Node s2=new Node(88);

Node s3=new Node(93);

Node s4=new Node(98);

l.siraliEkle(s);

l.siraliEkle(s1);

l.siraliEkle(s2);

l.siraliEkle(s3);

l.siraliEkle(s4);

l.yazdir();

l.yazdir();

l.siraliEkle(83);

l.yazdir();}

public void siraliEkle(int i) { } }

**4)Sondan bir öncekini silme.**

package çözüm;

public class Node {

int veri;

Node geri;

Node ileri;

Node(int veri) {

this.veri = veri;

this.ileri = null;

this.geri = null; }}

public class CbL {

Node bas;

public void ekle(int deger) {

Node yeni= new Node(deger);

if (bas == null) {

bas = yeni;

return; }

Node m = bas;

while (m.ileri!= null) {

m = m.ileri; }

m.ileri= yeni;

yeni.geri=m }

public void sondanBirOncekiSil() {

if (bas == null || bas.ileri == null) {

System.***out***.println("Silinecek düğüm yok.");

return; }

Node m = bas;

while (m.ileri != null) {

m= m.ileri; }

Node sondanOnceki = m.geri;

if (sondanOnceki != null) {

sondanOnceki.ileri = null; }}

public void yazdir() {

Node m = bas;

while (m != null) {

System.***out***.print(m.veri + " ");

m = m.ileri; }

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

public static void main(String[] args) {

CbL I= new CbL();

I.ekle(10);

I.ekle(20);

I.ekle(30);

I.ekle(40);

System.***out***.println("Başlangıç listesi:");

I.yazdir();

I.sondanBirOncekiSil();

System.***out***.println("Sondan bir önceki düğüm silindikten sonra:");

I.yazdir(); } }

**5)Tersten yazma.**

package çözüm;

public class Node {

int yeni;

Node ileri;

Node (int yeni) {

this.yeni = yeni;

this.ileri = null; }}

public class TyBl {

Node bas;

public void ekle(int veri) {

Node yeniDugum = new Node(veri);

if (bas == null) {

bas = yeniDugum; }

else {

Node m= bas;

while (m.ileri != null) {

m = m.ileri; }

m.ileri = yeniDugum; }}

public TyBl tersCevir() {

TyBl yeniListe = new TyBl();

Node m = bas;

while (m!= null) {

Node yeniDugum = new Node(m.yeni);

yeniDugum.ileri = yeniListe.bas;

yeniListe.bas = yeniDugum;

m = m.ileri; }

return yeniListe; }

public void yazdir() {

Node m = bas;

while (m != null) {

System.***out***.print(m.yeni + " ");

m = m.ileri; }

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

public static void main(String[] args) {

TyBl l = new TyBl();

l.ekle(10);

l.ekle(20);

l.ekle(30);

l.ekle(40);

System.***out***.println("Orijinal liste:");

l.yazdir();

TyBl tersListe = l.tersCevir();

System.***out***.println("Ters çevrilmiş liste:");

tersListe.yazdir();

}

}