Soru 1)

class Ogrenci {

String isim;

int notu;

public Ogrenci(String isim, int notu) {

this.isim = isim;

this.notu = notu;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Ogrenci ogrenci;

OgrenciListesi sonraki;

public OgrenciListesi(Ogrenci ogrenci) {

this.ogrenci = ogrenci;

this.sonraki = null;

}

public void ogrenciEkle(Ogrenci yeniOgrenci) {

OgrenciListesi yeniOgrenciNode = new OgrenciListesi(yeniOgrenci);

OgrenciListesi temp = this;

while (temp.sonraki != null) {

temp = temp.sonraki;

}

temp.sonraki = yeniOgrenciNode;

}

public Ogrenci enBasariliOgrenci() {

if (this.sonraki == null) {

return null;

}

OgrenciListesi temp = this.sonraki;

Ogrenci enBasarili = temp.ogrenci;

while (temp != null) {

if (temp.ogrenci.notu > enBasarili.notu) {

enBasarili = temp.ogrenci;

}

temp = temp.sonraki;

}

return enBasarili;

}

}

Ogrenci ogrenci1 = new Ogrenci("Ali", 85);

Ogrenci ogrenci2 = new Ogrenci("Ayşe", 92);

Ogrenci ogrenci3 = new Ogrenci("Mehmet", 78);

OgrenciListesi ogrenciListesi = new OgrenciListesi(ogrenci1);

ogrenciListesi.ogrenciEkle(ogrenci2);

ogrenciListesi.ogrenciEkle(ogrenci3);

Ogrenci enBasarili = ogrenciListesi.enBasariliOgrenci();

if (enBasarili != null) {

System.***out***.println("En Başarılı Öğrenci:");

System.***out***.println("İsim: " + enBasarili.isim);

System.***out***.println("Not: " + enBasarili.notu);

} else {

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

}

}

}

SORU 2)

class Kisi {

String isim;

double kilo;

double boy;

double vke;

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

this.isim = isim;

this.kilo = kilo;

this.boy = boy;

this.vke = hesaplaVke(kilo, boy);

}

private double hesaplaVke(double kilo, double boy) {

return kilo / (boy \* boy);

}

public boolean obezMi() {

return vke >= 30;

}

public double getVke() {

return vke;

}

public void yazdir() System.out.println("İsim: " + isim + " | Kilo: " + kilo + " | Boy: " + boy + " | VKE: " + vke);

}

}

public class Main {

public static void main(String[] args) {

class KisiListesi {

Kisi kisi;

KisiListesi sonraki;

public KisiListesi(Kisi kisi) {

this.kisi = kisi;

this.sonraki = null;

}

public void sonaEkle(Kisi yeniKisi) {

KisiListesi yeniNode = new KisiListesi(yeniKisi);

KisiListesi temp = this;

while (temp.sonraki != null) {

temp = temp.sonraki;

}

temp.sonraki = yeniNode;

}

public void yazdir() {

KisiListesi temp = this;

while (temp != null) {

temp.kisi.yazdir();

temp = temp.sonraki;

}

}

public void obezleriBul() {

KisiListesi temp = this;

boolean obezVar = false;

while (temp != null) {

if (temp.kisi.obezMi()) {

System.out.println("Obez Kişi: " + temp.kisi.isim);

obezVar = true;

}

temp = temp.sonraki;

}

if (!obezVar) {

System.out.println("Obez kişi bulunmamaktadır.");

}

}

public void sil(String isim) {

KisiListesi temp = this;

KisiListesi onceki = null;

if (temp != null && temp.kisi.isim.equals(isim)) {

this.kisi = temp.sonraki.kisi;

this.sonraki = temp.sonraki.sonraki;

return;

}

while (temp != null && !temp.kisi.isim.equals(isim)) {

onceki = temp;

temp = temp.sonraki;

}

if (temp == null) return;

onceki.sonraki = temp.sonraki;

}

}

public class Main {

public static void main(String[] args) {

Kisi kisi1 = new Kisi("Ayşe", 45, 1.70);

Kisi kisi2 = new Kisi("Ali", 78, 1.90);

Kisi kisi3 = new Kisi("Ahmet", 56, 1.50);

Kisi kisi4 = new Kisi("Veli", 65, 1.69);

Kisi kisi5 = new Kisi("Kemal", 59, 1.63);

KisiListesi kisiListesi = new KisiListesi(kisi1);

kisiListesi.sonaEkle(kisi2);

kisiListesi.sonaEkle(kisi3);

kisiListesi.sonaEkle(kisi4);

kisiListesi.sonaEkle(kisi5);

System.***out***.println("Tüm Kişiler ve VKE:");

kisiListesi.yazdir();

System.***out***.println("\nObez Kişiler:");

kisiListesi.obezleriBul();

System.***out***.println("\nKemal'i listeden silme:");

kisiListesi.sil("Kemal");

kisiListesi.yazdir();

}

}

SORU 3)

class Node {

int veri;

Node onceki;

Node sonraki;

public Node(int veri) {

this.veri = veri;

this.onceki = null;

this.sonraki = null;

}

}

class DoublyLinkedList {

Node baslangic;

Node son;

public DoublyLinkedList() {

baslangic = null;

son = null;

}

public void siraliEkle(int yeniDeger) {

Node yeniNode = new Node(yeniDeger);

if (baslangic == null) {

baslangic = yeniNode;

son = yeniNode;

} else {

Node temp = baslangic;

if (yeniDeger < baslangic.veri) {

yeniNode.sonraki = baslangic;

baslangic.onceki = yeniNode;

baslangic = yeniNode;

}

else if (yeniDeger > son.veri) {

son.sonraki = yeniNode;

yeniNode.onceki = son;

son = yeniNode;

}

else {

while (temp != null && temp.veri < yeniDeger) {

temp = temp.sonraki;

}

yeniNode.onceki = temp.onceki;

yeniNode.sonraki = temp;

if (temp.onceki != null) {

temp.onceki.sonraki = yeniNode;

}

temp.onceki = yeniNode;

}

}

}

public void yazdir() {

Node temp = baslangic;

while (temp != null) {

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

temp = temp.sonraki;

}

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

}

public class Main {

public static void main(String[] args) {

DoublyLinkedList liste = new DoublyLinkedList();

liste.siraliEkle(69);

liste.siraliEkle(80);

liste.siraliEkle(88);

liste.siraliEkle(93);

liste.siraliEkle(98);

System.out.println("Başlangıçtaki Liste:");

liste.yazdir();

liste.siraliEkle(83);

System.out.println("83 Değeri Eklenmiş Liste:");

liste.yazdir();

}

}

}

SORU4)

class Node {

int veri;

Node onceki;

Node sonraki;

public Node(int veri) {

this.veri = veri;

this.onceki = null;

this.sonraki = null;

class DoublyLinkedList {

Node baslangic;

Node son;

public DoublyLinkedList() {

baslangic = null;

son = null;

}

public void sonaEkle(int yeniDeger) {

Node yeniNode = new Node(yeniDeger);

if (baslangic == null) {

baslangic = yeniNode;

son = yeniNode;

} else {

son.sonraki = yeniNode;

yeniNode.onceki = son;

son = yeniNode;

}

}

public void sondanBirOncekiSil() {

if (baslangic == null || baslangic.sonraki == null) {

System.out.println("Liste boş veya yalnızca bir düğüm var. Silme işlemi yapılamaz.");

return;

}

Node temp = son.onceki;

if (temp.onceki == null) {

baslangic = son;

} else {

temp.onceki.sonraki = son;

son.onceki = temp.onceki;

}

System.out.println("Sondan bir önceki düğüm silindi: " + temp.veri);

}

public void yazdir() {

if (baslangic == null) {

System.out.println("Liste boş.");

return;

}

Node temp = baslangic;

while (temp != null) {

System.out.print(temp.veri + " ");

temp = temp.sonraki;

}

System.out.println();

}

}

}

}

public class Main {

public static void main(String[] args) {

DoublyLinkedList liste = new DoublyLinkedList();

liste.sonaEkle(10);

liste.sonaEkle(20);

liste.sonaEkle(30);

liste.sonaEkle(40);

liste.sonaEkle(50);

System.***out***.println("Başlangıçtaki Liste:");

liste.yazdir();

liste.sondanBirOncekiSil();

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

liste.yazdir();

}

}

SORU 5)

class Node {

int veri;

Node sonraki;

public Node(int veri) {

this.veri = veri;

this.sonraki = null;

}

}

class SinglyLinkedList {

Node baslangic;

public SinglyLinkedList() {

baslangic = null;

}

public void sonaEkle(int yeniDeger) {

Node yeniNode = new Node(yeniDeger);

if (baslangic == null) {

baslangic = yeniNode;

} else {

Node temp = baslangic;

while (temp.sonraki != null) {

temp = temp.sonraki;

}

temp.sonraki = yeniNode;

}

}

public SinglyLinkedList terstenAktar() {

SinglyLinkedList yeniListe = new SinglyLinkedList();

Node temp = baslangic;

while (temp != null) {

Node yeniNode = new Node(temp.veri);

yeniNode.sonraki = yeniListe.baslangic;

yeniListe.baslangic = yeniNode;

temp = temp.sonraki;

}

return yeniListe;

}

public void yazdir() {

Node temp = baslangic;

while (temp != null) {

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

temp = temp.sonraki;

}

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

}

}

public class Main {

public static void main(String[] args) {

SinglyLinkedList liste = new SinglyLinkedList();

liste.sonaEkle(10);

liste.sonaEkle(20);

liste.sonaEkle(30);

liste.sonaEkle(40);

liste.sonaEkle(50);

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

liste.yazdir();

SinglyLinkedList terstenListe = liste.terstenAktar();

System.***out***.println("Tersten Aktarılmış Liste:");

terstenListe.yazdir();

}

}