

## NESNEYE DAYALI PROGRAMLAMA VİZE CEVAPLARI

```
interface ParasalIslemler {
    public int paraKazan();
}

abstract class Sporcu implements ParasalIslemler {
    String adSoyad;
    int yas;
    public Sporcu(String as, int y) {
        adSoyad=as;
        yas=y;
    }

    public String getAdSoyad() {
        return adSoyad;
    }

    public int getYas() {
        return yas;
    }

    public void setAdSoyad(String yeni) {
        adSoyad=yeni;
    }

    public void setYas(int yeni) {
        yas=yeni;
    }

    public String toString() {
        return "Ad Soyad=" +adSoyad + " Yas=" + yas;
    }

    public abstract int paraKazan();
}
```

```
abstract class Atlet extends Sporcu {
    int maas;
    public Atlet(String as, int y, int m) {
        super(as,y);
        maas=m;
    }

    public int getMaas() {
        return maas;
    }

    public void setMaas(int yeni) {
        maas=yeni;
    }

    public String toString() {
        return super.toString()+ "\n" + " Maas=" + maas;
    }

    public abstract int paraKazan();
}
```

```

class Kosucu extends Atlet {
    private String mesafe;

    public Kosucu(String as, int y, int maas, String m) {
        super(as, y, maas);
        mesafe=m;
    }

    public String getMesafe() {
        return mesafe;
    }

    public void setMesafe(String m) {
        mesafe=m;
    }

    public String toString() {
        return super.toString()+ " Mesafe=" +mesafe;
    }

    public int paraKazan() {
        return maas;
    }
}

```

```

class Basketbolcu extends Sporcu {
    int sayiOrt;
    int asistOrt;
    int katsayi;
    Mevki mevki;

    public Basketbolcu(String as, int y, int so, int ao, int k, int secim) {
        super(as,y);
        sayiOrt=so;
        asistOrt=ao;
        katsayi=k;
        setMevki(secim);
    }

    public int getSayiOrt() {
        return sayiOrt;
    }

    public int getAsistOrt() {
        return asistOrt;
    }

    public int getKatsayi() {
        return katsayi;
    }

    public Mevki getMevki() {
        return mevki;
    }

    public void setSayiOrt(int yeni) {
        sayiOrt=yeni;
    }
}

```

```

public void setAsistOrt(int yeni) {
    asistOrt=yeni;
}

public void setKatsayi(int yeni) {
    katsayi=yeni;
}

public void setMevki(int secim) {
    switch (secim)
    {
        case 1:
            mevki=Mevki.PIVOT;
            break;
        case 2:
            mevki=Mevki.GUARD;
            break;
        case 3:
            mevki=Mevki.FORVET;
            break;
        default:
            mevki=Mevki.FORVET;
    }
}

public String toString() {
    return super.toString()+ "\n" +
        " Sayı Ort="+sayiOrt+" Asist Ort="+asistOrt+" Katsayi="+katsayi + "Mevki=" +
        getMevki().toString();
}

public int paraKazan() {
    return (sayiOrt+asistOrt)*katsayi;
}

static class Mevki {
    public static final Mevki PIVOT=new Mevki("Pivot");
    public static final Mevki GUARD=new Mevki("Guard");
    public static final Mevki FORVET=new Mevki("Forvet");

    private String etiket;

    public Mevki(String e) {
        etiket=e;
    }

    public String getEtiket() {
        return etiket;
    }

    public String toString() {
        return etiket;
    }
}
}

```

```

import java.util.*;
class SporKulubu implements Cloneable {
    private String kulupIsmi;
    private String kulupBaskani;
    private String sehir;
    private Vector sporcular;

    public SporKulubu(String i, String b, String s, Vector v) {
        kulupIsmi=i;
        kulupBaskani=b;
        sehir=s;
        sporcular=new Vector();
        sporcular=v;
    }

    public String getKulupIsmi() {
        return kulupIsmi;
    }

    public String getKulupBaskani() {
        return kulupBaskani;
    }

    public String getSehir() {
        return sehir;
    }

    public Vector getSporcular() {
        return sporcular;
    }

    public void setKulupIsmi(String yeni) {
        kulupIsmi=yeni;
    }

    public void setKulupBaskani(String yeni) {
        kulupBaskani=yeni;
    }

    public void setSehir(String yeni) {
        sehir=yeni;
    }

    public void setSporcular(Vector yeni) {
        sporcular=yeni;
    }

    public void yeniSporcuEkle(Sporcu yenitransfer) {
        sporcular.addElement(yenitransfer);
    }

    public String toString() {
        String temp=new String("");

        temp=temp+"KulupIsmi=" + kulupIsmi + " Kulup Baskani=" + kulupBaskani + " Sehir=" +
        sehir + "\n";
        temp=temp+"Sporcu Listesi:" + "\n";
        for (int i=0; i<sporcular.size(); i++) {
            temp=temp+sporcular.elementAt(i).toString();
            temp=temp+"\n";
        }
    }
}

```

```

    }
    return temp;
}

public Object clone() {
    try {
        SporKulubu sporObj = (SporKulubu) super.clone();
        sporObj.sporcular=(Vector)sporcular.clone();
// DERSTE POLYGON ÖRNEĞİNDE ANLATILDIĞI GİBİ, DEEP CLONE İŞLEMİ YAPILMALIDIR.
// EĞER DEEP CLONE YAPILMADI İSE NEDEN TERCİH EDİLMEDİĞİ KONUSUNDA YORUM
// YAZILMALIDIR.
        return sporObj;
    }
    catch(CloneNotSupportedException e) {
        return null;
    }
}

}

import java.util.*;
public class SporTest {

    public static void main(String[] args) {
        // Bosluk Doldurma Sorusunun Cevaplari
        Sporcu s1[]=new Sporcu[10]; // DOGRU

        // Sporcu s2=new Atlet("Semra Aksu",40,1000); // YANLIS - Compile Hatasi - Abstract Class
        Cannot Be Instantiated

        Sporcu s3=new Basketbolcu("Cenk Akyol",20,10,3,1500,3); // DOGRU

        // Sporcu s4=new Kosucu("Semra Aksu",40,1000,"100m"); // YANLIS - Compile Dogru Fakat
        RunTime Hatasi
        // Basketbolcu b1=(Basketbolcu)s4; // ClassCastException olur.

        Atlet a1=new Kosucu("Semra Aksu",40,1000,"100m"); // DOGRU
        Kosucu k2=(Kosucu)a1;
        System.out.println(k2.getMesafe());

        Sporcu s5=k2; // DOGRU
        // System.out.println(s5.getMesafe()); // Metot cagrimi yanlis, 2 asagidaki gibi downcast
        edilmeli.
        System.out.println(s5.paraKazan());

        //Atlet a2=new Kosucu("Semra Aksu",40,1000,"100m"); // HATA-Compile Etmez.
        // Basketbolcu b2=(Basketbolcu)a2; // Inconvertible types

        ParasalIslemler p1=k2; // DOGRU

        ParasalIslemler p2=k2; // DOGRU
        System.out.println(((Kosucu)p2).getMesafe());

        // Bosluk Doldurma Sorusunun Cevaplari Sonu

```

```

// Clone kullanımı örnekler
Kosucu k3=new Kosucu("Semra Aksu",40,1000,"100m");
Kosucu k4=(Kosucu) k3.clone();

System.out.println(k3.toString());
System.out.println(k4.toString());

Vector v1=new Vector();
v1.addElement(k3);
v1.addElement(k4);
SporKulubu spork1=new SporKulubu("KSK","Ersezgin","Izmir",v1);
System.out.println(spork1);

SporKulubu spork2=(SporKulubu) spork1.clone();
System.out.println(spork2);
// Clone kullanımı sonu

// Clone kullanimini asagida polymorphic isleyisi gosterdikten sonra
// programinizda mevcut olan nesnelerle de gosterebilirsiniz.
// Yukaridaki gibi ayri nesneler olusturmaniza gerek yoktur.

// Polymorphic isleyis ve “downcasting” örneklenmesi.
Kosucu kos1=new Kosucu("Metin Tekin",45,1000,"Maraton");
Kosucu kos2=new Kosucu("Elvan A",25,1000,"5000m");
Basketbolcu basket1=new Basketbolcu("Kaya Peker",28,14,5,2500,1);
Vector sporcuvektor=new Vector();
sporcuvektor.addElement(kos1);
sporcuvektor.addElement(kos2);
sporcuvektor.addElement(basket1);
SporKulubu spork=new SporKulubu("Besiktas","Demiroren","Istanbul",sporcuvektor);

for (int i=0; i<spork.getSporcular().size();i++) {
    Sporcu sporcu1=(Sporcu) spork.getSporcular().elementAt(i);
    System.out.println(sporcu1.toString());
    System.out.println(sporcu1.paraKazan());

    if (sporcu1 instanceof Basketbolcu) {
// AŞAĞIDAKİ SATIRDAKİ GİBİ “DOWNCASTING” YAPILMALIDIR. 5 puan
        if (((Basketbolcu) sporcu1).getMevki()).getEtiket()=="Pivot") {
            int eskikatsayi=((Basketbolcu)sporcu1).getKatsayi();
            int yenikatsayi=eskikatsayi*2;
            ((Basketbolcu)sporcu1).setKatsayi(yenikatsayi);
        }
    }

    if (sporcu1 instanceof Kosucu) {
//AŞAĞIDAKİ SATIRDAKİ GİBİ DOWNCASTING YAPILMALIDIR. 5 puan
        if (((Kosucu)sporcu1).getMesafe()=="Maraton") {
            int eskimaas=((Kosucu)sporcu1).getMaas();
            int yenimaas=eskimaas+1500;
            ((Kosucu)sporcu1).setMaas(yenimaas);
        }
    }

    System.out.println(sporcu1.paraKazan());
}

System.out.println("Main End");
}
}

```