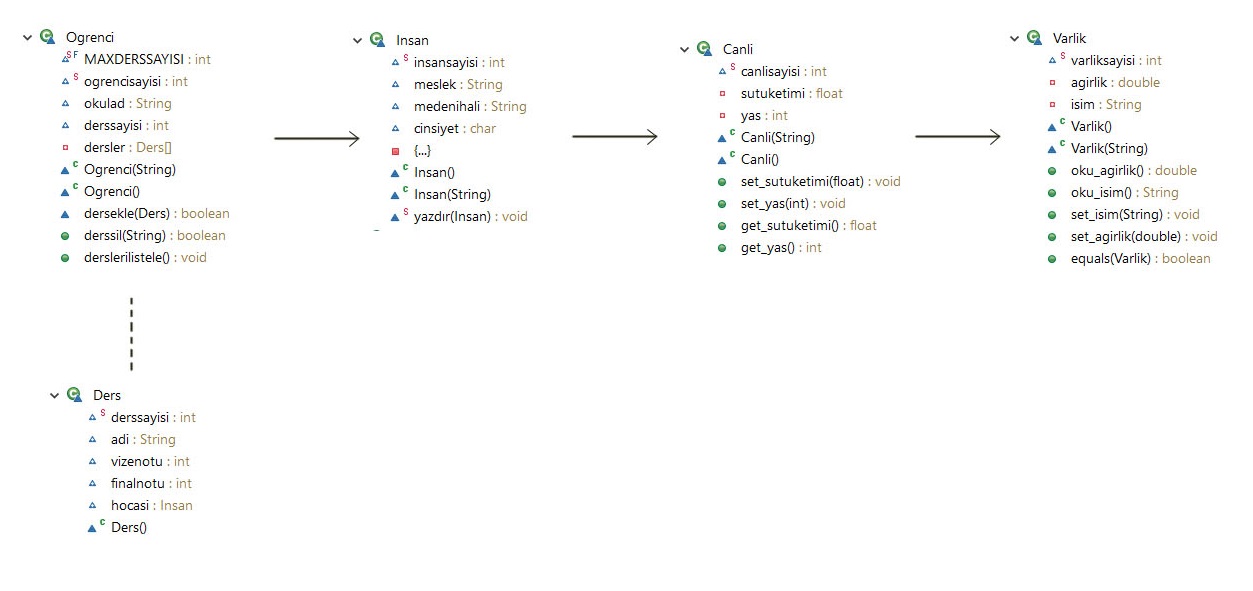
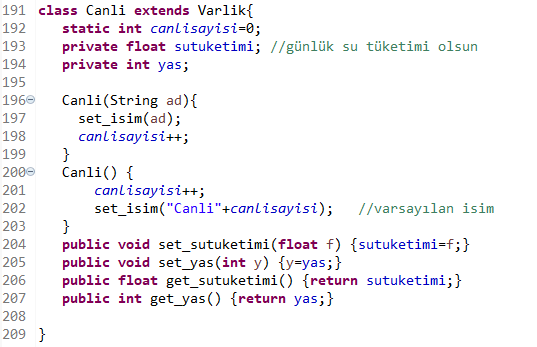
LAB5 Cevapları

1)



2)

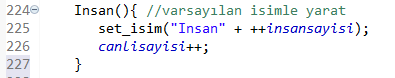


3) Varsayılan ilklendirici ve parametreli ilklendirici ile ilklendirmeler yapılmıştır.  
Evet.

4) Olmaz. Çünkü her öğrenci oluşturulduğunda ders instansı oluşuyor. Dolayısıyla o1.dersekle(d1) demeden de d1.hocasi=hoca ataması yapılabilir.

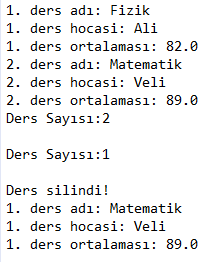
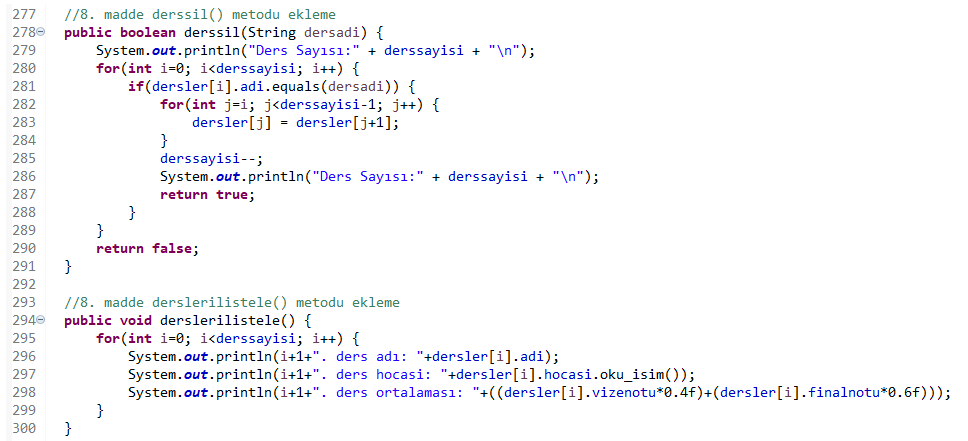
5) o1 ve o2 'de ortak olan 'd1' dersi new ile yeniden oluşturulmadığından iki objede tek bir objeyi refere ettiğinden 2 ders yerine 1 ders yazılmış oluyor.

6) i) Hata olmaz. Çünkü çocuk sınıflar varsayılan olarak super() gizli metodu ile atayı çağırır.  
ii)



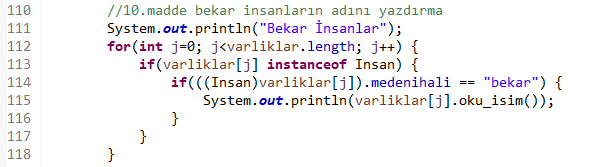
7) i) Çünkü Varlik() sınıfı "yas" değişkenine sahip değil fakat onun çocuğu sahip olduğundan cast yaptık.  
ii) Çünkü Varlik() sınıfı "oku\_isim()" değişkenine kendisi sahip olduğu için cast yapmadık.

8)



9) Nesneyi özelden genele olacak şekilde, ataya çocuğu atarsak zaten çocuk atanın özelliklerini kapsadığı için gizli cast oluşur. Dolayısıyla bu sonucu yeni oluşturulan çocuğa cast ederek atayabiliriz.

10)



11) equals() metodu değer veya değerlerin eşitlik kontrolünü yapıp, true veya false döndürür.  
Varlik sınıfındaki overload edilmiş equals() metodu, isim (string) karşılaştırması yapar ve değerler eşit ise true aksi halde false döndürür.  
equals() metodu varsayılan olarak tüm objelerde bulunur, override edilebilir.

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** lab5;

*/\*\**

*\**

*\* @author zeki*

*\*/*

**class** **test** {

*/\*\**

*\* @param args the command line arguments*

*\*/*

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

*/\*\*\*\*\*BOLUM 1: her sÄ±nÄ±ftan birer adet instanslar yaratalÄ±m ve geliÅŸigüzel bilgiler girelim\*\*\*\*/*

*/\*\*\*\*\*BOLUM 1.1: VarlÄ±k,Canli, Insan sÄ±nÄ±flarÄ±na ait instanslar yaratalÄ±m\*\*\*\*\*/*

Varlik v = **new** Varlik("insan");

v.set\_agirlik(72.5);

Canli c = **new** Canli("kedi");

c.set\_sutuketimi(0.3f);

c.set\_yas(1);

Insan i= **new** Insan();

i.medenihali="bekar";

i.meslek="ogretmen";

i.set\_sutuketimi(2);

i.set\_yas(30);

*/\*\*\*\*BOLUM 1.2:bir öÄŸrenci yaratalÄ±m ve tüm bilgilerini girelim\*\*\*\*\*/*

Ogrenci o1= **new** Ogrenci("ali");

o1.set\_agirlik(70); *//aÄŸÄ±rlÄ±k özelliÄŸi korunuyor..kapsüllenmiÅŸ*

o1.set\_yas(20); *//yas özelliÄŸi korunmuyor..kapsüllenmemiÅŸ.*

o1.set\_sutuketimi(2.5f);

o1.cinsiyet='K';

o1.medenihali="bekar";

o1.okulad="Mersin Univ.";

Ders d1= **new** Ders(); *//dersi yaratÄ±p,bilgiler girip, o1' e ekleyelim*

d1.adi="Nesneye Yonelik Programlama";

d1.vizenotu=30;

d1.finalnotu=60;

Insan hoca = **new** Insan(); *//hocayÄ± yaratÄ±p, d1 ' in hocasÄ± olarak sonra atayalÄ±m*

o1.dersekle(d1); *//d1 i, o1 e ekliyoruz*

hoca.medenihali="evli";

hoca.cinsiyet='E';

hoca.set\_yas(50);

hoca.set\_sutuketimi(1.2f);

d1.hocasi = hoca; *// hoca'yÄ± d1' e burada atadÄ±k. hoca ve d1.hocasi aynÄ± nesne oldu*

*/\*\*\*\*BOLUM 1.3: varsayÄ±lan bilgilerle yeni bir öÄŸrenci yaratalÄ±m. d1 dersini ve iki yeni dersi alsÄ±n\*\*\*\*/*

Ogrenci o2= **new** Ogrenci();

o2.dersekle(d1);

o2.dersekle(**new** Ders()); *//yeni bir ders daha ekleyelim. Ders bilgilerini sonra deÄŸiÅŸtirebiliriz*

o2.dersekle(**new** Ders()); *//yeni bir ders daha ekleyelim. Ders bilgilerini sonra deÄŸiÅŸtirebiliriz*

*/\*\*\*\*\*\*BOLUM 2:statik ve instans tanÄ±mlÄ± sayÄ±cÄ±larÄ±n deÄŸerlerini analiz edelim \*\*\*\*\*\*\*\*\*\*\*/*

*//static deÄŸiÅŸkenlere hem sÄ±nÄ±f ismiyle hemde instans ismiyle ulaÅŸabiliriz.*

System.out.println("\nBolum 2 ............................");

System.out.print("varlik sayisi:"+Varlik.varliksayisi+ ", canli sayisi:"+ c.canlisayisi+", insan sayisi:"+i.insansayisi);

System.out.print(" ogrenci sayisi:"+ Ogrenci.ogrencisayisi);

System.out.print(" ders sayisi:"+ Ders.derssayisi);

System.out.println(" o1.ders\_sayisi:"+ o1.derssayisi+" o2.ders\_sayisi:"+o2.derssayisi);

*/\*\*\*\*\*\*BOLUM 3:nesneler arasÄ± cast iÅŸlemini analiz edelim \*\*\*\*\*\*\*\*\*\*\*/*

*/\*\*\*BOLUM 3.1: bir nesneyi atasÄ±na cast edelim, sonra da atasÄ±ndan kendisine cast edelim\*\*\*\*/*

System.out.println("\nBolum 3.1 ............................");

Insan i1= o1; *//o1 öÄŸrencisiyle aynÄ± özelliklere sahip bir insan yaratmanÄ±n kÄ±sayolu : gizli cast*

*// nesneyi atasÄ±na cast ettik = özelden genele geçiÅŸ = her ögrenci bir insandÄ±r--> açÄ±k caste gerek yok*

*//i1 artÄ±k bir öÄŸrenci instansÄ±na sahip oldu.*

Insan.yazdÄ±r(i1);

Ogrenci o3=**null**, o4=**null**;

**if** (i1 **instanceof** Ogrenci)

o3 = (Ogrenci)i1; *//i1 Ogrenci instansÄ±na sahip olduÄŸundan mümkün. yoksa hata çÄ±kar,*

**if** (i **instanceof** Ogrenci)

o4 = (Ogrenci)i; *//i Ogrenci instansÄ±na sahip deÄŸil, bu satÄ±ra gelmez,*

System.out.println("o3="+o3+", o4="+o4); *//adreslerin string ifadesini yazdÄ±r*

*/\*\*\*BOLUM 3.2: nesneleri en büyük atalarÄ±na cast edelim sonra castle ilgili iÅŸlemler yapalÄ±m\*\*\*/*

System.out.println("\nBolum 3.2 ............................");

Varlik [] varliklar = **new** Varlik[5]; *// 5 varlÄ±k yaratalÄ±m. yukarda yarattÄ±klarÄ±mÄ±zÄ± sÄ±rayla atayalÄ±m*

varliklar[0] = v;

varliklar[1] = c;

varliklar[2] = i;

varliklar[3] = o1;

varliklar[4] = o2;

*//canli olan varlÄ±klarÄ±n adÄ±nÄ± ve yaÅŸÄ±nÄ± yazdÄ±ralÄ±m*

**for** (int j=0;j<varliklar.length;j++)

**if** (varliklar[j] **instanceof** Canli)

System.out.println(varliklar[j].oku\_isim()+"--> yaÅŸ:"+((Canli)varliklar[j]).get\_yas());

*//10.madde bekar insanlarÄ±n adÄ±nÄ± yazdÄ±rma*

System.out.println("Bekar Ä°nsanlar");

**for**(int j=0; j<varliklar.length; j++) {

**if**(varliklar[j] **instanceof** Insan) {

**if**(((Insan)varliklar[j]).medenihali == "bekar") {

System.out.println(varliklar[j].oku\_isim());

}

}

}

*/\*\*\*BOLUM 4: nesnelerin esitligi ile ilgili iÅŸlemler yapalÄ±m\*\*\*/*

*//equals metodu Object sinifinda zaten tanimli...tanimi soyle Object.equals (Object o)*

*//fakat biz Varlik sinifinda equals metodunu overload ediyoruz*

System.out.println("\nBolum 4 ............................");

System.out.println("o1 ve o2 esit mi (bizim equal metodu ile): "+ o1.equals(o2));

Ogrenci o5= **new** Ogrenci (o1.oku\_isim());

System.out.println("o1 ve o5 esit mi(bizim equal metodu ile): "+ o1.equals(o5));

System.out.println("o1 ve o5 esit mi(varsayilan Oject.equal metodu ile): "+ o1.equals((Object)o5));

o1=o5;

System.out.println("o1=o5 atamasÄ±ndan sonra (varsayilan Oject.equal metodu ile): "+ o1.equals((Object)o5) + "\n");

*//Ders ekleyip silme (Sonradan eklendi!)*

Ogrenci ogr1 = **new** Ogrenci();

Ders ders1 = **new** Ders();

ders1.adi = "Fizik";

ders1.vizenotu = 70;

ders1.finalnotu = 90;

Insan hoca1 = **new** Insan("Ali");

ders1.hocasi = hoca1;

ogr1.dersekle(ders1);

Ders ders2 = **new** Ders();

ders2.adi = "Matematik";

ders2.vizenotu = 80;

ders2.finalnotu = 95;

Insan hoca2 = **new** Insan("Veli");

ders2.hocasi = hoca2;

ogr1.dersekle(ders2);

ogr1.derslerilistele();

boolean derssilindi = ogr1.derssil("Fizik");

**if**(derssilindi) {

System.out.println("Ders silindi!");

ogr1.derslerilistele();

}

**else** {

System.out.println("Ders bulunamadÄ±!");

}

}

}

**class** **Varlik** {

**static** int varliksayisi=0;

**private** double agirlik;

**private** String isim;

Varlik(){ *//varsayÄ±lan isimle yarat*

varliksayisi++;

isim="Varlik"+varliksayisi;

}

Varlik(String ad){ *//verilen isimle yarat*

isim=ad;

varliksayisi++;

}

**public** double oku\_agirlik(){ **return** agirlik; }

**public** String oku\_isim(){**return** isim;}

**public** void set\_isim(String s) {isim=s;}

**public** void set\_agirlik(double kg) {agirlik=kg;}

**public** boolean equals (Varlik a){

**return** a.isim.equals(isim);

}

}

**class** **Canli** **extends** Varlik{

**static** int canlisayisi=0;

**private** float sutuketimi; *//günlük su tüketimi olsun*

**private** int yas;

Canli(String ad){

set\_isim(ad);

canlisayisi++;

}

Canli() {

canlisayisi++;

set\_isim("Canli"+canlisayisi); *//varsayÄ±lan isim*

}

**public** void set\_sutuketimi(float f) {sutuketimi=f;}

**public** void set\_yas(int y) {y=yas;}

**public** float get\_sutuketimi() {**return** sutuketimi;}

**public** int get\_yas() {**return** yas;}

}

*//Insan: varsayÄ±lan olarak 1 yaÅŸÄ±nda 2.5 kg, adÄ± "Insanxxx" olan erkek bir bebek olsun*

**class** **Insan** **extends** Canli{

**static** int insansayisi=0;

String meslek="yok"; *//en basit ilklendirme yolu = tanÄ±mlanÄ±rken ilklendirme*

String medenihali="bekar"; *//evli, bekar, dul*

char cinsiyet = 'E'; *//E(rkek) ve K(Ä±z)*

{*//instans ilklendirici, bazÄ± ilklendirmeleri burada da yapabiliriz yada yapÄ±landÄ±rÄ±cÄ± da yapabiliriz*

set\_agirlik(2.5);

set\_sutuketimi(0.2f);

set\_yas(1);

}

Insan(){ *//varsayÄ±lan isimle yarat*

set\_isim("Insan" + ++insansayisi);

canlisayisi++;

}

*//super("Insan" + ++insansayisi); //supersÄ±nÄ±fÄ±n (atasÄ±nÄ±n) ilgili yapÄ±landÄ±rÄ±cÄ±sÄ±nÄ± (bu durumda String parametreli olan) iÅŸletir.*

*//yazÄ±lmazsa yapÄ±landÄ±rÄ±cÄ±nÄ±n ilk satÄ±rÄ±nda parametresiz gizli bir super(); çaÄŸÄ±rÄ±mÄ± vardÄ±r.*

Insan(String ad) { *//verilen isimle yarat*

**super**(); *//super() zaten gizli olarak var..yazmamÄ±za gerek yoktu.*

set\_isim(ad);

++insansayisi;

}

**static** void yazdÄ±r(Insan i) { *//verilen insana ait bilgileri yazdÄ±rsÄ±n*

System.out.println("adÄ±:"+ i.oku\_isim());

System.out.println("agirlik:"+ i.oku\_agirlik());

System.out.println("su tüketimi:"+ i.get\_sutuketimi());

System.out.println("yas:"+ i.get\_yas());

System.out.println("cinsiyet:"+ i.cinsiyet);

System.out.println("meslek:"+ i.meslek);

}

}

**class** **Ogrenci** **extends** Insan {

**static** **final** int MAXDERSSAYISI=20; *//bu sÄ±nÄ±f altÄ±nda global constant*

**static** int ogrencisayisi=0;

String okulad; *//instans deÄŸiÅŸkenleri otomatik null(sÄ±fÄ±r)lanÄ±r, varsayÄ±lan "" deÄŸerini görürsünüz*

int derssayisi=0; *//ilklendirme yapmasakta otomatik sifir olurdu*

**private** Ders [] dersler;

Ogrenci(String ad) {

**super**(ad);

ogrencisayisi++;

meslek="ogrenci";

dersler= **new** Ders[MAXDERSSAYISI];

}

Ogrenci() {

**super**("Ogrenci" + ++ogrencisayisi);

meslek="ogrenci";

dersler= **new** Ders[MAXDERSSAYISI];

}

*//ders basarili eklenirse true yoksa false donsun. dersler MAXDERSSAYISINI asmasin*

boolean dersekle(Ders d) {

**if** (derssayisi<MAXDERSSAYISI) {

dersler[derssayisi]=d;

derssayisi++;

**return** **true**; *//burada exit olsun*

}

**return** **false**; *//buraya gelmisse maxderssayisina ulasilmis demektir*

}

*//8. madde derssil() metodu ekleme*

**public** boolean derssil(String dersadi) {

System.out.println("Ders SayÄ±sÄ±:" + derssayisi + "\n");

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

**if**(dersler[i].adi.equals(dersadi)) {

**for**(int j=i; j<derssayisi-1; j++) {

dersler[j] = dersler[j+1];

}

derssayisi--;

System.out.println("Ders SayÄ±sÄ±:" + derssayisi + "\n");

**return** **true**;

}

}

**return** **false**;

}

*//8. madde derslerilistele() metodu ekleme*

**public** void derslerilistele() {

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

System.out.println(i+1+". ders adÄ±: "+dersler[i].adi);

System.out.println(i+1+". ders hocasi: "+dersler[i].hocasi.oku\_isim());

System.out.println(i+1+". ders ortalamasÄ±: "+((dersler[i].vizenotu\*0.4f)+(dersler[i].finalnotu\*0.6f)));

}

}

}

**class** **Ders** {

**static** int derssayisi=0;

String adi;

int vizenotu;

int finalnotu;

Insan hocasi;

Ders(){

derssayisi++;

adi="Ders"+derssayisi; *//varsayÄ±lan ders adÄ±*

}

}