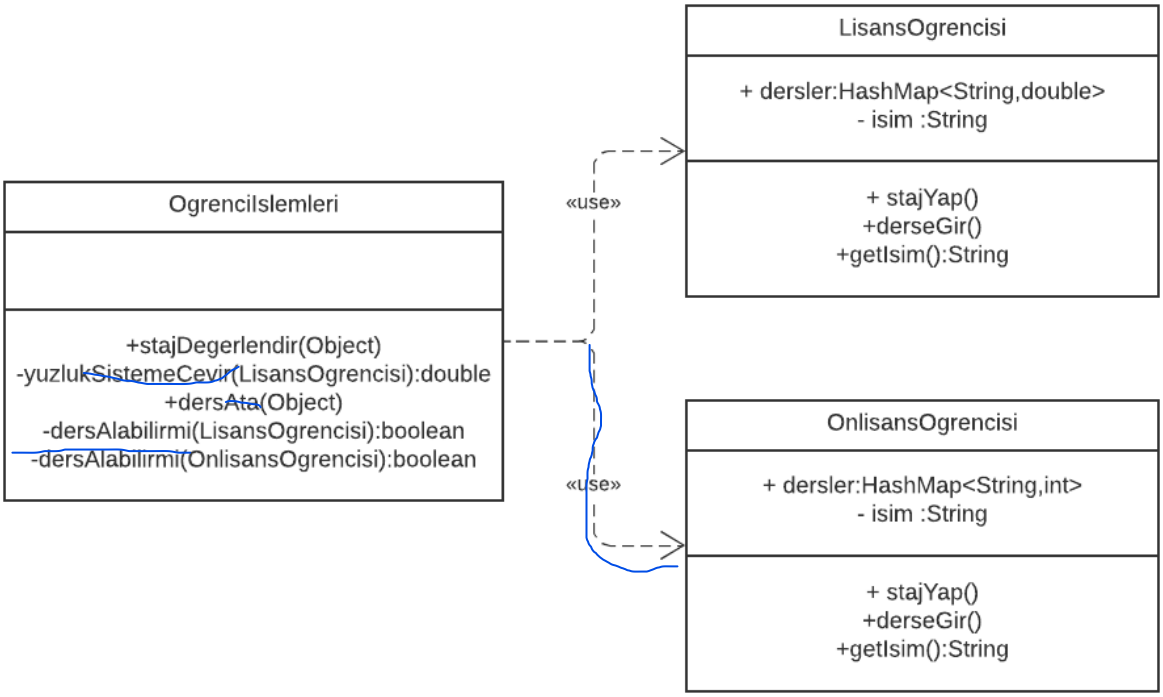



```java
public class OgrenciIsleri {
    public void stajDegerlendir(Object object) {
            if (object instanceof LisansOgrencisi) {
                    LisansOgrencisi ogrenci = (LisansOgrencisi) object;
                    ogrenci.dersler.put("Staj", Double.valueOf(4));
            }else if (object instanceof OnlisansOgrencisi) {
                    OnlisansOgrencisi ogrenci = (OnlisansOgrencisi) object;
                    ogrenci.dersler.put("Staj", Integer.valueOf(100));
            }
    }
    private int yuzlukSistemeCevir(double deger) {
            return (int)(deger-2)*20+60;
    }
    public void dersAta(Object object,String ders) {
            if (object instanceof LisansOgrencisi) {
                    LisansOgrencisi ogrenci = (LisansOgrencisi) object;
                    if (dersAlabilirmi(ogrenci)) {
                            ogrenci.dersler.put(ders, Double.valueOf(0));
                    }
            }else if (object instanceof OnlisansOgrencisi) {
                    OnlisansOgrencisi ogrenci = (OnlisansOgrencisi) object;
                    if (dersAlabilirmi(ogrenci)) {
                            ogrenci.dersler.put(ders, Integer.valueOf(0));
                    }
            }
    }
    private boolean dersAlabilirmi(LisansOgrencisi ogrenci) {
            double ortalama=0;
            for(Map.Entry<String, Double> entry :ogrenci.dersler.entrySet()) {
                ortalama += entry.getValue().doubleValue();
            }
            ortalama /=ogrenci.dersler.size();
            return yuzlukSistemeCevir(ortalama)>60;
    }
    private boolean dersAlabilirmi(OnlisansOgrencisi ogrenci) {
            int ortalama=0;
            for(Map.Entry<String, Integer> entry :ogrenci.dersler.entrySet()) {
                ortalama += entry.getValue().intValue();
            }
            ortalama /=ogrenci.dersler.size();
            return ortalama>60;
    }
}
```

LisansOgrencisi sınıfında 4 lük not sistemi kullanılırken, OnlisansOgrencisi sınıfında ise 100 lük not sistemi kullanılmaktadır.

1.Soru	2.Soru	3.Soru	4.Soru

SORU 1. Yukarıda verilen UML de verilen sınıf tasarımındaki problemleri hangi prensibe ya da prensiplere uygun olmadığını da belirterek anlatınız ve problemlerin çözümü için bir tasarım öneriniz. **(40p)**

1. Single principle ilkesi ihlal edilmiş OgrenciIsleri sınıfında Çünkü staj değerlendirme ve ders atama gibi 1 den fazla sorumluluk verilmiş. Bu sınıf Bölünmeli

Üstteki dersAta snf Örenci(Neden örenci yüksek lisan m yoksa ön lisans m bilmiyoruz bu yüzden örenci diye tanmladk

Burda dersAlabilirmi alyor ise ortalamaHesapla > 60 dan büyük olanlar döndürüyoruz

```
public class KayitIslemleri {
	public void dersAta(Ogrenci ogrenci,String ders) {
		if (dersAlabilirmi(ogrenci)) {
			ogrenci.dersEkle(ders, Double.valueOf(0));
		}
	}
	private boolean dersAlabilirmi(Ogrenci ogrenci) {
		return ogrenci.ortalamaHesapla()>60;
	}
}

public class StajIslemleri {
	public void stajDegerlendir(Ogrenci ogrenci) {
		ogrenci.dersEkle("Staj", Double.valueOf(4));
	}
}
```

Staj Deerlendir. Burada ekstra biey yok

2. Open close prensibine göre yeni bir öğrenci türü eklenemez eklenirse OgrenciIsleri sınıfı değiştirilmeli. Bu durumu çözmek için Ogrenci Interfacesi yazmak lazım.

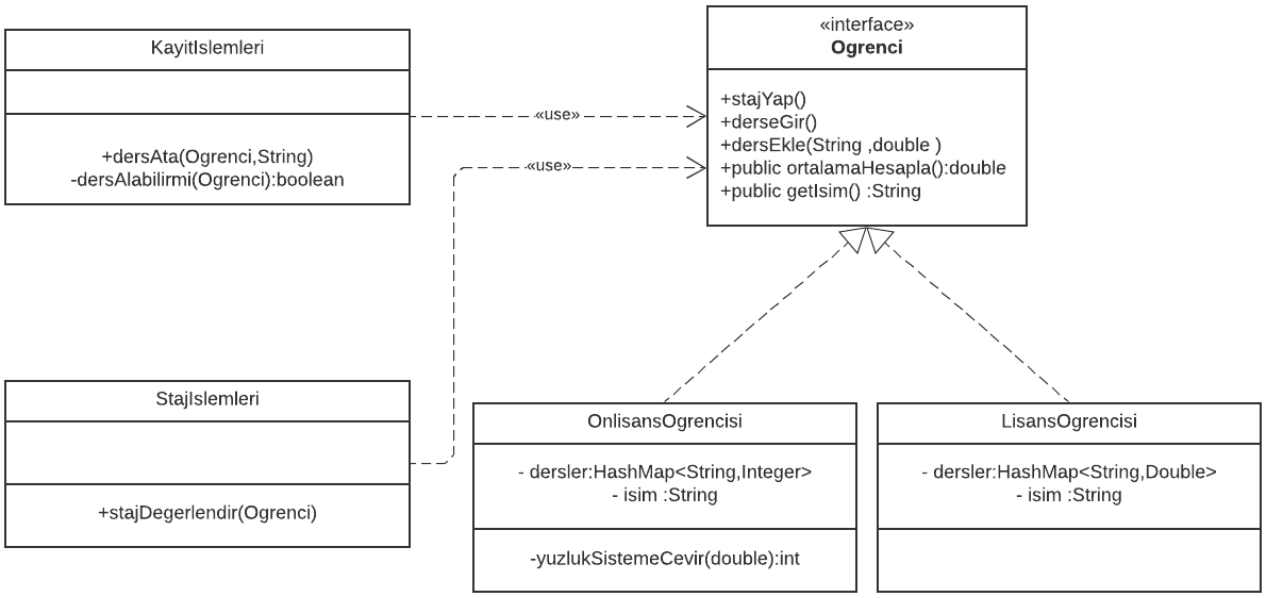
Örenci leri ilemleri Orenci adnda bir interface oluturuyoruz bu örenci ilerindeki methodlar Ogrenciye atyoruz

Bu hashmap uml de belirtiyor

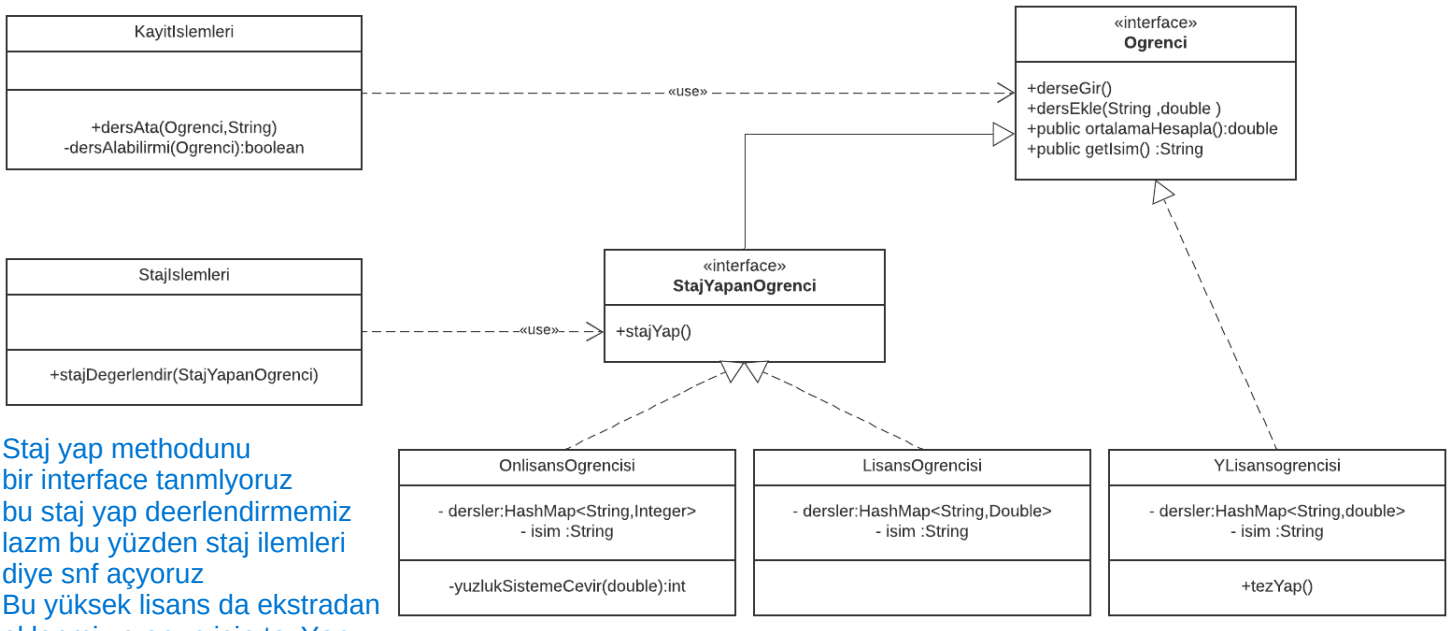
```
public interface Ogrenci {
	public void stajYap() ;
	public void derseGir() ;
	public void dersEkle(String ders,double not) ;
	public double ortalamaHesapla();
	public String getIsim() ;
}public class LisansOgrencisi implements Ogrenci{

	private HashMap<String,Double> dersler = new HashMap<>();
	private String isim;
	public void stajYap() {
		System.out.println("staj yapıldı");
	}
	public void derseGir() {
		System.out.println("Lisans dersine girildi");
	}
	public String getIsim() {
		return isim;
	}
	@Override
	public void dersEkle(String ders, double not) {
		dersler.put(ders, Double.valueOf(not));

	}
	@Override
	public double ortalamaHesapla() {
		double ortalama=0;
		for(Map.Entry<String, Double> entry :dersler.entrySet()) {
		    ortalama += entry.getValue().doubleValue();
		}
		ortalama /=dersler.size();
		return ortalama;
	}
}

public class OnlisansOgrencisi implements Ogrenci{
	public HashMap<String,Integer> dersler = new HashMap<>();
	private String isim;
	public void stajYap() {
		System.out.println("staj yapıldı");
	}
	public void derseGir() {
		System.out.println("Lisans dersine girildi");
	}
	public String getIsim() {
		return isim;
	}
	@Override
	public void dersEkle(String ders, double not) {
		dersler.put("Staj", yuzlukSistemeCevir(Double.valueOf(100)));
```

```
        }
        private int yuzlukSistemeCevir(double deger) {
                return (int)(deger-2)*20+60;
        }
        @Override
        public double ortalamaHesapla() {
                double ortalama=0;
                for(Map.Entry<String, Integer> entry :dersler.entrySet()) {
                    ortalama += entry.getValue().intValue();
                }
                ortalama /=dersler.size();
                return yuzlukSistemeCevir(ortalama);
        }
}
```

3. Dependency inversion ilkesine göre üst seviye sınıf detaylara bağlı olamamalı ama OgrenciIsleri sınıfı 100 lük sistem mi kullanılıyor 4 lük sistem mi kullanılıyor bilmesi gerekiyor. Buda soyutlamaya zarar veriyor.
    - Bu yüzden ortalama hesaplama ve yüzlük sisteme dönüştürme metotları öğrenci sınıflarına taşındı
4. Kapsülleme açısından problemli buda kodu kırılgan yapar.
    - Öğrencilerin dersler değişkeni private yapıldı ve dersGir metodu eklendi



SORU 2. Aşağıdaki sınıf UML diyagramında verilen tasarıma eklendiğinde liskov yerine geçme prensibini de ihlal etmeyecek yeni bir tasarım gerçekleştiriniz. **(25p)**



Staj yap methodunu
bir interface tanmlyoruz
bu staj yap deerlendirmemiz
lazm bu yüzden staj ilemleri
diye snf açyoruz
Bu yüksek lisans da ekstradan
eklenmi ve onun için tezYap

1.Soru	2.Soru	3.Soru	4.Soru

SORU 3. Nesne tabanlı programlama özelliklerini soru 2 de yaptığınız tasarımdan örnekleyerek kazanımları ve amacını anlatınız. **(20p).**

SORU 4. Singleton tasarım kalıbı hakkında bilgi veriniz ve bir singleton nesnenin clonunu almak mümkünmüdür açıklayınız. (değil ise neden mümkün ise hangi durumlarda açıklanmalı) **(15p).**

Bir nesnenin yalnzca bir örnei oluturulmas salanr
Bir snftan kaç nesne üretildiini kontrol etmek için kullanlr
Singleton clonunu alnabilir ama singleton olmaktan çkyor

Soru 3 :
Çok biçimlilik(polymorphism) : staj yap methodunu birden fazla yerde kullanyoruz .
Kapsülleme – encapsulation: tezYap () Amaç ile ilgili bütün ilemler bu snfta gerçekletirilmeli

Inteherince : Oklar ile gösteriliyor bir snfn dier snf miras ald

Abstraction :