

Mühendislik Fakültesi-Nesne Tabanlı Programlama Dersi Ara Sınav Cevapları

- 1) Nesne tabanlı programlamada kullanılan 4 temel prensibin ismini (İngilizce veya Türkçe) yazın. (8p)
- 2) Nesne tabanlı programlamada kullanılan aşağıdaki iki kavramı kısaca açıklayınız. (12p)

C1)	Prensip
1	Abstraction / Soyutlama
2	Inheritance / Miras
3	Encapsulation / Kapsülleme
4	Polymorphism / Çok biçimlilik

C2)
Method overloading
Adaş metotlar: Aynı isimde aynı/benzer işlevi yerine getiren ancak argüman listesi farklı olan metotlar tanımlanmasına verilen addır. Java dilinde doğrudan desteklenir, python dilinde method overloading mekanizması daha farklıdır.
Method overriding
Metot aşırı yükleme/Metot güncelleme: Miras alan child/sub sınıfın parent/super sınıftan miras yolu ile aldığı metotları değiştirmesi/güncellemesi.

- 3) Aşağıdaki Python/Java koduna göre tabloda verilen sorular için uygun method_ismi/kavramı yazınız. (8p)

```
class soru:
    def __init__(self):
        pass

    def __del__(self):
        pass
```

```
x = soru()
```

```
class soru{
    public soru(String str){
    }

    protected void finalize(){
    }
}

class Main {
    public static void main(String[] args) {
        soru x = new soru("test");
    }
}
```

C3)	
Class?	soru
Object?	x
Constructor?	__init__(self) #python soru() #java
Destructor?	__del__(self) #python finalize() #java

- 4) Aşağıda aynı programın Python / Java kodu verilmiştir. Program çalıştığı zaman oluşacak ekran çıktısını tam ve eksiksiz olarak yazınız (12p)

<pre>class animal: def __init__(self, name, code): self.name=name self.code=code print("Animal oluşturuldu") class reptile(animal): def __init__(self, name, code, kind): super().__init__(name, code) self.kind=kind print("reptile oluşturuldu") self.str() def str(self): print(str(self.code)+" "+self.name+" "+self.kind) kobra= reptile("Kobra",5,"zehirli")</pre>	<pre>class animal{ String name; int code; public animal(String name, int code){ this.name=name; this.code=code; System.out.println("Animal oluşturuldu"); } } class reptile extends animal{ String kind; public reptile(String name, int code, String kind){ super(name, code); this.kind=kind; System.out.println("reptile oluşturuldu"); this.str(); } public void str(){ System.out.println(this.code+" "+this.name+" "+this.kind); } } class Main { public static void main(String[] args) { reptile kobra = new reptile("Kobra",5,"zehirli"); } }</pre>
---	--

C4)

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
Animal oluşturuldu
reptile oluşturuldu
5 kobra zehirli
```

- Doğrudan sub sınıftan (reptile) nesne üretmeye çalışıyoruz, sub sınıfın miras yolu ile türediği için (animal sınıfından) önce süper sınıf çağrılır, dolayısı ile önce süper sınıfın yapılandırıcısı çağrılır.
- Sonra sub sınıfın yapılandırıcısı çağrılır, sub sınıf name ve code atama işlemi için süper sınıfının yapılandırıcısını çağırıyor, kind atama işlemi ise kendisi yapıyor, ardından str() methodunu çağırıyor.
- *Miras söz konusu olduğunda çalışma sıralarına dikkat edilmesi gerekiyor.*

5)

Bir tamsayı dizisini {0-1} aralığına dönüştürme işlemi normalizasyon yöntemlerinden bir tanesidir.(örneğin [0 50 100] dizisi [0 0.5 1] dizine dönüşecek) buna göre verilen bir diziyi normalize ederek geri döndüren programı aşağıda ki kurallara uygun şekilde nesne tabanlı olarak kodlayınız (35p).

- normalize edilecek dizi sınıfın yapılandırıcı metoduna parametre olarak verilecek. (5p)
- normalize_et() metodu bu giriş dizisini normalize edecek. Bu metod yapılandırıcı metod içinden çağrılacak.(15p)
- Hem normalize edilmemiş, hem de normalize edilmiş diziler ayrı sınıf değişkenlerinde saklanacak. (5p)
- Sınıfın yok edici metodu her iki diziyi de silecek. (5p)
- Sınıf yapısının şablon olarak doğru olması gerekmektedir. (5p)

C5)

Program aşağıdaki NTP programlanabilir, normalize edilecek dizi yapılandırıcıya gönderilmiş, normalize_et() metodu yapılandırıcı içinden çağrılmış, normalize_et() methodu diziyi normalize etmiş, her iki dizi sınıf değişkeni olarak saklanmış.

Normalize etme algoritması şu şekilde çalışır, dizideki en büyük ve en küçük sayı bulunur. Dizi de her sayıdan en küçük sayı çıkarılarak en büyük-en küçük farkına bölünür.

```
class soru5:
    def __init__(self, dizi):
        self.dizi=dizi
        print(self.dizi)
        self.normalize_et()

    def normalize_et(self):
        minimum=min(self.dizi)
        maximum=max(self.dizi)
        self.normalized=[(i-minimum)/(maximum-minimum) for i in self.dizi]
        print(self.normalized)

    def bul(self,indis):
        try:
            print("orjinal değer:",self.dizi[indis])
            print("normalize edilmiş:",self.normalized[indis])
        except:
            print("Bir hata oluştu")

x= soru5([1,3,5,7,9])

x.bul(3)
x.bul("a")
x.bul(-14)
x.bul(114)
x.bul(3.2)
```

C6)

istisna oluşturabilecek durumlar.

- İndis olarak rakam yerine karakter/string verilebilir.
- Negatif indis verilebilir (Pythonda negatif indis verilebilir örneğin -1 dizinin son elemanını gösterir, istisna oluşturmaz ancak -100 istisna oluşturur)
- Dizi boyutundan daha büyük indis verilebilir.
- İndis olarak tam sayı yerine double/float verilebilir.
- B şıkkı yukarıdaki gibi kodlanabilir.
- x.bul(3) şeklinde çağrıldığında istisna oluşmadan çalışır, ancak diğerlerinin hepsinde istisna oluşur ekrana "Bir hata oluştu" mesajını yazar.

Önemli Not: kodlama sorularına 1-1 aynı şekilde çözmüş olma şartı beklenemez. Yaklaşımınız ve kod doğru ise tam puan alırsınız.