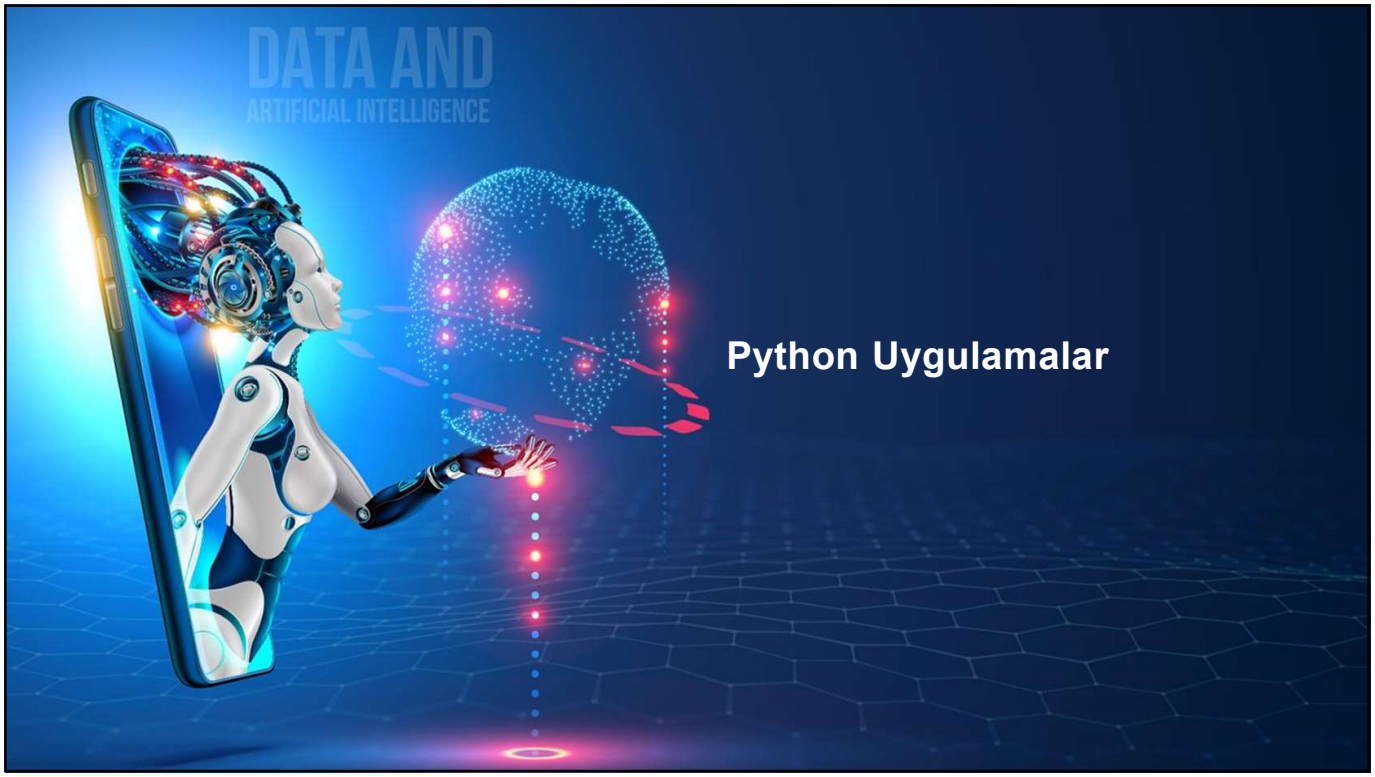

Python Uygulamalar

Nesne Tabanlı Programlama

1- OgrenciNotları adında bir sınıf tanımlayın. İki niteliği olsun öğrenci adı-soyadı ve notlar. Ayrıca öğrencinin geçip/kaldığını gösteren durumu adında fonksiyon da olsun. Geçme notu 60.

Kod

```
class OgrenciNotları():
    def __init__(self,ad_soyad,notu):
        self.ad_soyad=ad_soyad
        self.notu=notu
    def durum(self):
        if self.notu<60:
            return "Kaldı"
        else:
            return "Geçti"
o1=OgrenciNotları("Ugur Ozcan",77)
o2=OgrenciNotları("Ali Can",55)
o1.durum()
o2.durum()
```

2- Diktörtgen adında bir sınıf tanımlayın ve alanını, çevresini ve köşegen uzunluğunu hesaplayın.

Kod

```
class Diktörtgen():
    def __init__(self,uzunluk,genişlik):
        self.uzunluk=uzunluk
        self.genişlik=genişlik
    def alanı(self):
        alan=self.uzunluk*self.genişlik
        return alan
    def çevresi(self):
        çevre=2*self.uzunluk+2*self.genişlik
        return çevre
    def köşegeni(self):
        köşegen=(self.uzunluk**2+self.genişlik**2)**0.5
        return köşegen
d1=Diktörtgen(3,4)
d1.alanı()
d1.çevresi()
d1.köşegeni()
```

3- Daire adında bir sınıf tanımlayın, yarıçapını alın, alanını ve çevresini hesaplayın.

Kod

```
class Üçgen():
    def __init__(self,yarıçap):
        self.yarıçap=yarıçap
    @staticmethod
    def pi():
        return 22/7
    def alanı(self):
        alan=Üçgen.pi()*self.yarıçap**2
        return alan
    def çevresi(self):
        çevre=2*Üçgen.pi()*self.yarıçap
        return çevre
u1=Üçgen(3)
u1.alanı()
u1.çevresi()
```

4- Bir tamsayıyı romen rakamına dönüştürmek için bir Python sınıfı yazın.

sayı=[1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1]
romen=["M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"]

Kod

```
class SayıToRomen():
    romen_to_deger=[1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1]
    deger_to_romen=["M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"]
    def __init__(self,sayı):
        self.sayı=sayı
        self.romen=""
    def cevir(self):
        i=0
        while self.sayı>0:
            for _ in range(self.sayı//self.romen_to_deger[i]):
                self.romen=self.romen+self.deger_to_romen[i]
                self.sayı=self.sayı-self.romen_to_deger[i]
            i=i+1
        return self.romen
ç1=SayıToRomen(300)
ç1.cevir()
```

5- Bir romen rakamını tamsayıya dönüştürmek için bir Python sınıfı yazın.

sayı=[1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1]
romen=["M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"]

Kod

```
class RomenToSayı():
    romen_to_deger=[1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1]
    deger_to_romen=["M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V",
"IV", "I"]
    def __init__(self,romen):
        self.romen=romen
        self.sayı=int()
    def romene_cevir(self):
        for i in self.romen:
            indeks=self.deger_to_romen.index(i)
            self.sayı=self.sayı+self.romen_to_deger[indeks]
        return self.sayı
ç1=RomenToSayı("MMMMCLXXXVI")
ç1.romene_cevir()
```

6- Farklı tamsayılardan oluşan bir tuple'ın olası bütün alt kümelerini elde etmek için bir Python sınıfı yazın.

Input : (4, 5, 6)
Output : {(), (4,), (4, 5), (4, 5, 6), (5,), (5, 6), (6,)}

Kod

```
class AltKüme():
    def __init__(self,tuplex):
        self.tuplex=tuplex
        self.altküme=set()
    def alt_küme(self):
        for i in range(0,len(self.tuplex)+1):
            for j in range(0,i+1):
                self.altküme.add(self.tuplex[j:i])
        return self.altküme
ç1=AltKüme((4,5,6))
ç1.alt_küme()
```

7- Toplamı belirli bir hedef sayıya eşit olan belirli bir diziden bir eleman çifti (iki sayının indeksi) bulmak için bir Python sınıfı yazın.

Her girdi için bir çözüm olacak ve aynı elemanı iki kez kullanmayın.

Input: numbers= [10,20,10,40,50,60,70], target=50
Output: 3, 0

Kod

```
class IndeksBul():
    def __init__(self,sayılar,hedef):
        self.sayılar=sayılar
        self.hedef=hedef
        self.indeks1=""
        self.indeks2=""
    def bul(self):
        for i in self.sayılar:
            for j in self.sayılar:
                if i+j==self.hedef:
                    self.indeks1=self.sayılar.index(i)
                    self.indeks2=self.sayılar.index(j)
                    break
        return self.indeks1,self.indeks2
IndeksBul([10,20,10,40,50,60,70],50).bul()
```

8- Bir dizi n gerçek sayıdan toplamı sıfıra eşit olan üç öğeyi bulmak için bir Python sınıfı yazın.

Input array : [-25, -10, -7, -3, 2, 4, 8, 10]
Output : [[-10, 2, 8], [-7, -3, 10]]

Kod

```
class SıfırıBul():
    def __init__(self,liste):
        self.liste=liste
        self.sıfırlar=[]
    def bul(self):
        for i in range(0,len(self.liste)-2):
            for j in range(i+1,len(self.liste)-1):
                for k in range(j+1,len(self.liste)):
                    if self.liste[i]+self.liste[j]+self.liste[k]==0:

self.sıfırlar.append([self.liste[i],self.liste[j],self.liste[k]])
        return self.sıfırlar
ç1=SıfırıBul([-25, -10, -7, -3, 2, 4, 8, 10])
ç1.bul()
```


DATA AND
ARTIFICIAL INTELLIGENCE
Teşekkürler