

1) Sınıflara Giriş

```
class Matematik:

    def kare_al(self,x):
        return x**2

    def us_al(self, taban, us):
        return taban**us

    def daire_alani(self,pi, r):
        return pi*r*r


a=Matematik()

print( a.daire_alani(3.14, 4) )

print(a.kare_al (5))

b=a.us_al(3,4)
print(b)
```

2) Yapılandırıcı metot ve Sınıf değişkeni kullanımı

```
class Matematik:

    def __init__(self):
        self.pi=3.14

    def kare_al(self,x):
        return x**2

    def us_al(self, taban, us):
        return taban**us

    def daire_alani(self,pi, r):
        return pi*r*r


a=Matematik()

print(a.pi)
```

3) Sınıf değişkeni kullanımı

```
class Matematik:

    def __init__(self):
        self.pi=3.14

    def kare_al(self,x):
        return x**2

    def us_al(self, taban, us):
        return taban**us

    def daire_alani(self,r):
        return self.pi*r*r

a=Matematik()

pi=2
print(pi)
print(a.pi)
print(a.daire_alani(2))
```

4) Yapılandırıcıya parametre aktarımı

```
class Matematik:

    def __init__(self,pi):
        self.pi=pi

    def kare_al(self,x):
        return x**2

    def us_al(self, taban, us):
        return taban**us

    def daire_alani(self,r):
        return self.pi*r*r

a=Matematik(3.14)

print(a.daire_alani(2))
```

5) NTP Basit bir sayısal loto örneği

```
import random

class Loto:

    def __init__(self):
        self.tutulanlar=[random.randint(1,50) for i in range(6)]

    def tutulanlari_yaz(self):
        print(self.tutulanlar)

    def tahminleri_yaz(self):
        print(self.tahminler)

    def tahminler(self, tahminler):
        self.tahminler=tahminler

    def tutanlar(self):
        dogrular=[]
        for x in self.tahminler:
            for y in self.tutulanlar:
                if x==y and x not in dogrular:
                    dogrular.append(x)

        if(len(dogrular)==0):
            print("0 adet tutan")
        else:
            print(len(dogrular)," adet tutan tahmin var",dogrular)

a=Loto()
tahminler=[3,7,8,4,3,3]
a.tutulanlari_yaz()

a.tahminler(tahminler)
a.tahminleri_yaz()
a.tutanlar()
```

```
[38, 36, 4, 48, 46, 48]
[3, 7, 8, 4, 3, 3]
1 adet tutan tahmin var [4]
```

6) Nesneyi silmek, ve yokedici metodun çalışması

```
import random

class Loto:

    def __init__(self):
        self.tutulanlar=[random.randint(1,50) for i in range(6)]

    def tutulanlari_yaz(self):
        print(self.tutulanlar)

    def tahminleri_yaz(self):
        print(self.tahminler)

    def tahminler(self, tahminler):
        self.tahminler=tahminler

    def tutanlar(self):
        dogrular=[]
        for x in self.tahminler:
            for y in self.tutulanlar:
                if x==y and x not in dogrular:
                    dogrular.append(x)

        if(len(dogrular)==0):
            print("0 adet tutan")
        else:
            print(len(dogrular)," adet tutan tahmin var",dogrular)

    def __del__(self):
        print("Nesne siliniyor")

a=Loto()
tahminler=[3,7,8,4,3,3]
a.tutulanlari_yaz()

a.tahminler(tahminler)
a.tahminleri_yaz()
a.tutanlar()

del(a)
```

```
[39, 50, 50, 35, 34, 46]
[3, 7, 8, 4, 3, 3]
0 adet tutan
Nesne siliniyor
```

7) Modul kullanımı

Bir önceki programı **ntp.py** ismiyle kaydettiğimizi varsayıyoruz.

```
import ntp

a=ntp.Loto()
tahminler=[3,7,8,4,3,3]
a.tutulanlari_yaz()

a.tahminler(tahminler)
a.tahminleri_yaz()
a.tutanlar()
```

Ekran görüntüsü: neden iki defa çalıştı ?

```
[4, 24, 46, 25, 14, 6]
[3, 7, 8, 4, 3, 3]
1 adet tutan tahmin var [4]
[29, 21, 24, 1, 28, 23]
[3, 7, 8, 4, 3, 3]
0 adet tutan
```