Bilgisayar Programlama 2

Ders Notu: 2

Sayma (Counting)

- Çoğu zaman programlarımızın bir şeyin kaç kez gerçekleştiğini saymasını isteriz. Örneğin, bir oyunda bir oyuncunun kaç tur attığının takip edilmesi gerekebilir veya bir matematik programının özel bir özelliğe sahip kaç sayı olduğunu sayması gerekebilir. Saymanın anahtarı, sayıyı tutmak için bir değişken kullanmaktır.
- **Örnek 1:** Bu program kullanıcıdan 10 sayı alır ve bu sayılardan kaç tanesinin 10'dan büyük olduğunu sayar.

```
1   count = 0
2
3   for i in range(10):
4       sayi = int(input("Bir sayi giriniz: "))
5       if sayi > 10:
6           count = count + 1
7
8   print("Ondan buyuk sayi adedi: ", count)
```

- Sayma son derece yaygın bir işlemdir. İki temel unsur vardır:

    1. count=0 Saymaya 0'dan başladığını ifade eder.
    2. count=count+1 Sayıyı 1 artırır.

Sayma (Counting)

- **Örnek 2:** Önceki örneğin bu modifikasyonu, kullanıcının girdiği sayılardan kaçının 10'dan büyük olduğu ve kaçının 0'a eşit olduğunu sayar. İki şeyi saymak için iki sayım değişkeni kullanırız.

```
1   count1 = 0
2   count2 = 0
3
4   for i in range(10):
5       sayi = int(input("Bir sayi giriniz: "))
6       if sayi > 10:
7           count1 = count1 + 1
8       if sayi == 0:
9           count2 = count2 + 1
10
11  print("Ondan buyuk sayi adedi: ", count1)
12  print("Sifira eşit sayi adedi: ", count2)
```

Sayma (Counting)

- **Örnek 3:** 1'den 100'e kadar olan sayılardan karelerinin kaçının 4 ile bittiğini sayar.

```
1   count = 0
2
3   for i in range(1,101):
4       kareSayi = i*i
5       if kareSayi % 10 == 4:
6           count = count + 1
7
8   print("1 den 100 e kadar olan sayi karelerinin sonu 4 ile bitenlerin sayisi:", count)
```

Toplama (Summing)

- Sayma işlemiyle yakından ilişkili olan toplama işleminde, bir dizi sayıyı toplamayı ifade eder.
- **Örnek 1:** 1'den 100'e kadar olan sayıların toplamı, çalışma şekli şöyledir: Her yeni sayıyla karşılaştığımızda, onu toplamımıza (s) ekleriz.

```
1   s = 0
2
3   for i in range(1,101):
4       s = s+i
5
6   print("Toplam = ", s)
```

Örnek 2: Kullanıcıdan 10 sayı isteyen ve ardından bunların ortalamasını hesaplayan uygulama:

```
1   s = 0
2
3   for i in range(10):
4       num = int(input("Lütfen bir sayi giriniz: "))
5       s = s+num
6
7   ort = s/10
8   print("Ortalama = ", ort)
```

Yer Değiştirme (Swapping)

- Çoğu zaman x ve y olmak üzere iki değişkenin değerlerini değiştirmek isteyeceğiz. Aşağıdakileri denemek çözüm için denenebilir,

```
x = y
y = x
```

- Ama bu işe yaramayacaktır. x = 3 ve y = 5 örnek değerleri için ilk satır x'i 5'e ayarlayacak, ama sonra ikinci satır da y'yi 5'e ayarlayacak çünkü x artık 5 tir. Buradaki önemli nokta, x'in değerini kaydetmek için üçüncü bir değişken kullanmaktır:

```
hold = x
x = y
y = hold
```

- Birçok programlama dilinde, değişkenleri değiştirmek için kullanılan yaygın yöntem budur. Ancak Python, bir kısayol sunar:

```
x,y = y,x
```

Bayrak Değişkenleri (Flag Variables)

- Bir flag değişkeni, programınızın bir bölümünün başka bir bölümünde bir şey olduğunda bunu bilmesini sağlamak için kullanılabilir.
- **Örnek:** Bir sayının asal olup olmadığını belirleyen uygulama.

```
1   sayi = int(input("Bir sayi giriniz= "))
2
3   flag = 0
4
5   for i in range(2,sayi):
6       if sayi % i == 0:
7           flag = 1
8
9   if flag == 1:
10      print("Asal degil")
11  else:
12      print("Asal sayi")
```

- Bir sayının asal sayı olması için 1 ve kendisinden başka böleni olmaması gerekir. Yukarıdaki programın çalışma şekli şöyledir: flag 0'dan başlar. Ardından 2'den sayi-1'e kadar döngü yapılır. Bu değerlerden biri bölen çıkarsa, flag 1'e ayarlanır. Döngü bittiğinde, bayrağın ayarlanıp ayarlanmadığını kontrol ederiz. Ayarlandıysa, bir bölen olduğunu ve sayının asal olmadığını biliyoruz. Aksi takdirde, sayı asal olmalıdır.

Maksimum ve Minimum Değerler (Maxes and Mins)

- Programlamada sıkça karşılaşılan bir görev, bir değer dizisindeki en büyük veya en küçük değeri bulmaktır.
- **Örnek:** Kullanıcıdan on pozitif sayı girmesini istediğimiz ve ardından en büyük sayıyı yazdırdığımız bir uygulama.

```
1   largest = int(input("Pozitif bir deger giriniz: "))
2
3   for i in range(9):
4       sayi = int(input("Pozitif bir deger giriniz: "))
5       if sayi > largest:
6           largest = sayi
7
8   print("En Büyük Sayi = ", largest)
```

- Buradaki kilit nokta, şimdiye kadar bulunan en büyük sayıyı takip eden `largest` değişkenidir. Bunu, kullanıcının ilk sayısına eşitleyerek başlıyoruz. Ardından, kullanıcıdan yeni bir sayı aldığımız her seferinde, kullanıcının sayısının mevcut en büyük değerden (`largest` değişkeninde saklanan) daha büyük olup olmadığını kontrol ediyoruz. Eğer daha büyükse, `largest` değişkenini kullanıcının sayısına eşitliyoruz.
- Bunun yerine en küçük değeri istiyorsak, gerekli tek değişiklik `>` işaretinin `<` olarak değiştirilmesidir.
- Listeler bölümünde, en büyük ve en küçük değerleri bulmanın daha kısa bir yolu incelenecektir.

Yorum Satırları (Comments)

- Yorum, programınızı okuyan birine gönderilen bir mesajdır. Yorumlar genellikle, özellikle karmaşık kod bölümlerinde, bir kod bölümünün ne yaptığını veya nasıl çalıştığını açıklamak için kullanılır. Yorumların programınız üzerinde hiçbir etkisi yoktur.
- **Tek satırlı yorum satırı:** Tek satırlık yorumlar için # karakterini kullanılır.

```
count = count + 2 # Sayacı 2 birim artırır.
```

- **Çok satırlı yorumlar:** Birden fazla satıra yayılan yorumlar için üçlü tırnak işareti kullanılır.

```
""" Program name": Hello world
Author: Brian Heinold
Date: 1/9/11 """

print('Hello world')
```

- Üçlü tırnak işaretlerinin güzel bir kullanım alanı, kodunuzun bazı bölümlerini yorum satırı haline getirmektir. Genellikle programınızı değiştirmek istersiniz ancak değişikliklerinizin işe yaramaması durumunda eski kodunuzu silmek istemezsiniz. Eski kodu yorum satırı haline getirerek, ihtiyaç duyduğunuzda hala orada olmasını sağlayabilir ve yeni programınız çalıştırıldığında göz ardı edilmesini sağlayabilirsiniz.

```
"""
print('Bu satir ve sonraki satir bir yorumun içinde.')
print('Bu satirlar çalistirilmayacak.')
"""
print('Bu satir bir yorumun içinde değil ve calistirilacak.')
```

Örnek Uygulama Pratikleri

- Kod okuyabilmek değerli bir beceridir. Bu bölümde, bazı basit programları inceleyip nasıl çalıştıklarını gözlemlenmektedir.
- **Örnek 1:** Aşağıdaki program, "Merhaba" kelimesini 5 ile 25 arasında rastgele bir sayıda yazdırır.

```
1   from random import randint
2
3   rastgeleSayi = randint(5,25)
4   for i in range(rastgeleSayi):
5       print('Hello')
```

Bir şeyi rastgele sayıda tekrarlamak için range(rastgeleSayi) kullanabiliriz. Ancak istersek, değişkeni atlayıp randint ifadesini doğrudan range fonksiyonuna koyabiliriz, aşağıda gösterildiği gibi.

```
1   from random import randint
2
3   for i in range(randint(5,25)):
4       print('Hello')
```

Örnek Uygulama Pratikleri

- **Örnek 2:** Aşağıdaki iki programı karşılaştırıldığında:

```
from random import randint

rand_num = randint(1,5)
for i in range(6):
    print('Hello '*rand_num)
```

```
Hello Hello
Hello Hello
Hello Hello
Hello Hello
Hello Hello
Hello Hello
```

```
from random import randint

for i in range(6):
    rand_num = randint(1,5)
    print('Hello '*rand_num)
```

```
Hello Hello Hello Hello
Hello Hello Hello Hello Hello
Hello Hello Hello
Hello
Hello Hello Hello Hello Hello
Hello
```

- Programlar arasındaki tek fark, rand_num ifadesinin yerleşimindedir. Birinci programda, rand_num ifadesi for döngüsünün dışında yer alır ve bu da rand_num'un programın başında bir kez ayarlanıp program biitene kadar aynı değeri koruduğu anlamına gelir. Dolayısıyla her print ifadesi aynı sayıda "Hello" yazdıracaktır. İkinci programda ise rand_num ifadesi döngünün içindedir. Her print ifadesinden hemen önce, rand_num'a yeni bir rastgele sayı atanır ve bu nedenle "Hello"nun yazdırılma sayısı satırdan satıra değişir.

Girinti (Indentation matters)

- Sık yapılan bir hata, yanlış girintilemedir.
- **Örnek**: 1 ile 100 arasında 10000 rastgele sayı üreten ve bunlardan kaç tanesinin 12'nin katı olduğunu sayan bir program yazalım.

```
from random import randint

count = 0
for i in range(10000):
    num = randint(1, 100)
    if num%12==0:
        count=count+1

print('12 nin katlarinin sayisi:', count)
```

```
1   from random import randint
2
3   count = 0
4   for i in range(10000):
5       num = randint(1, 100)
6       if num%12==0:
7           count=count+1
8
9       print('12 nin katlarinin sayisi:', count)
```

- Yukarıdaki örneği ele alıp son satırı girintili hale getirdiğimizi varsayalım. Program yine de çalışacak, ancak beklendiği gibi çalışmayacaktır. Çalıştırdığımızda, bir çok sayı çıktısı verir. Bunun nedeni, print ifadesini girintileyerek onu for döngüsünün bir parçası haline getirmemizdir, bu nedenle print ifadesi 10.000 kez çalıştırılacaktır.
- print ifadesini aşağıdaki gibi bir adım daha girintiliyoruz.

```
1   from random import randint
2
3   count = 0
4   for i in range(10000):
5       num = randint(1, 100)
6       if num%12==0:
7           count=count+1
8           print('12 nin katlarinin sayisi:', count)
```

Bu durumda sadece for döngüsünün bir parçası değil, aynı zamanda if ifadesinin de bir parçasıdır. Her yeni 12'nin katını bulduğumuzda, sayıyı yazdıracağız. Ne bu, ne de önceki örnek, istediğimiz şey değildir. Sadece programın sonunda sayıyı bir kez yazdırmak istiyoruz, bu yüzden print ifadesinin hiç girintili olmasını istemiyoruz.

Alıştırmalar

- **Örnek 1:** 1 ile 100 arasındaki sayıların karelerinin kaçının 1 ile bittiğini sayan bir program yazınız.
- **Örnek 2:** 1'den 100'e kadar olan sayıların karelerinin kaçının 4 ile, kaçının 9 ile bittiğini sayan bir program yazınız.
- **Örnek 3:** Kullanıcıdan n değerini girmesini isteyen ve ardından aşağıdaki hesaplamayı yapan bir program yazınız. ln fonksiyonu, matematik modülündeki log fonksiyonudur.

  $(1 + 1/2 + 1/3 + \ldots + 1/n) - ln(n)$

- **Örnek 4**: $1 - 2 + 3 - 4 + \cdots + 1999 - 2000$ toplamını hesaplayan bir program yazınız.
- **Örnek 5:** Kullanıcıdan bir sayı girmesini isteyen ve bu sayının bölenlerinin toplamını yazdıran bir program yazınız.
- **Örnek 6:** Bir sayı, kendisi hariç tüm bölenlerinin toplamına eşitse, mükemmel sayı olarak adlandırılır. Örneğin, 6 mükemmel bir sayıdır çünkü 6'nın bölenleri 1, 2, 3, 6'dır ve 6 = 1 + 2 + 3'tür. Başka bir örnek olarak, 28 mükemmel bir sayıdır çünkü bölenleri 1, 2, 4, 7, 14, 28'dir ve 28 = 1 + 2 + 4 + 7 + 14'tür. Ancak, 15 mükemmel bir sayı değildir çünkü bölenleri 1, 3, 5, 15'tir ve 15 = 1 + 3 + 5'tir. 10000'den küçük olan dört mükemmel sayıyı bulan bir program yazınız.
- **Örnek 7:** x, y ve z olmak üzere üç değişkenin değerlerini değiştiren bir program yazın; yani x, y'nin değerini, y, z'nin değerini ve z, x'in değerini alsın.
- **Örnek 8:** Kullanıcıdan 1 ile 10 arasında rastgele bir sayı tahmin etmesini isteyen bir program yazın. Doğru tahminde 10 puan eklenir, yanlış tahminde ise 1 puan kaybeder. Kullanıcıya tahmin etmesi için beş sayı isteyin ve tüm tahminler bittikten sonra puanını yazdırınız.