

Bilgisayar Programlama 2-2022/2023 Güz YY. Dönem Projesi

Rabia Şevval AYDIN

Kullanılan tüm fonksiyonlar aşağıdaki kodun genel görünümünü gösteren ekran görüntüsünde görülebilir. Açıklama sırasında değişkenler mavi, fonksiyonlar yeşil gösterilmiştir.

Kodun Açıklanışı Özet:

- A) Dosyadan veri çekme
- B) Salonunun ve temel değişkenlerin oluşturulması
- C) Ana menü ve temel fonksiyonları
- D) Kategoriyeye göre rezervasyon işlemi
- E) Ücret bilgilerinin hesaplanması

Kodun Genel Görünümü:

```
proje2.py 2 ...
1 > def rezervasyon(salon): #kategori ve bilet bilgisini alır, ilgili rezervasyon fonksiyonu ile ücretHesapla fonksiyonunu çağırır...
43
44 > def salonYazdır(salon): #salonu bastırır...
49
50 > def yeniEtkinlik(salon): #salonu boşaltıp tekrarda boş koltuklar ile doldurur...
59
60 > def rezervKategori_1(salon, biletNo): #1. kategori için rezervasyon işlemini yapar...
104
105 > def rezervKategori_2(salon, biletNo): #2. kategori için rezervasyon işlemini yapar...
199
200 > def rezervKategori_3(salon, biletNo): #3. kategori için rezervasyon işlemini yapar...
244
245 > def rezervKategori_4(salon, biletNo): #4. kategori için rezervasyon işlemini yapar...
344
345 > def ciroHesaplama(): #kategoriler için ayrı ve toplam ciroyu bastırır...
352
353 > def ücretHesapla(biletSayisi,kategoriNumarasi): #ücret,indirim,net tutar bilgilerini hesaplar...
417
418 > with open("indirim.txt","r") as dosya: #indirim dosyasından verileri çeker...
448
449
450 """#-----ANA MENÜ-----"""
451
452 salon = []
453 for i in range(0, 20): # salonu oluşturur
454     salon.append("-----")
455 kalanBilet=[100,100,100,100]
456 toplamCiro=[0,0,0,0]
457
458 girilen = 5
459 while girilen != 0:
460     girilen = input("*****\n** ANA MENÜ **\n1. Rezervasyon\n2. Salonu Yazdır\n3. Yeni Etkinlik\n4. Toplam Ciro\n0. ""Çıkış\n*****\nSeçiminiz: ")
461     match girilen:
462         case "1":
463             rezervasyon(salon)
464         case "2":
465             salonYazdır(salon)
466         case "3":
467             yeniEtkinlik(salon)
468         case "4":
469
Ln 418, Col 76 Spaces: 4 UTF-8 CRLF Python 3.10.5
```

Programın Çalışma Sürecinin Ayrıntılı Açıklanması:

A) DOSYADAN VERİ ÇEKME İŞLEMİ

Program çalışmaya 418. satırdan başlar. Önce indirim dosyasından verileri alır:

- **biletMax**: Maksimum bilet sayısını tutar.
- **ücretler[]**: Kategorilere göre bilet fiyatını tutar. 0.indiste 1.kategorinin fiyatı, 1.indiste 2.kategorinin fiyatı gibi.
- **liste1[]**: 1. kategorinin indirim bilgilerini tutar
- **liste2[]**: 2. kategorinin indirim bilgilerini tutar
- **liste3[]**: 3. kategorinin indirim bilgilerini tutar
- **liste4[]**: 4. kategorinin indirim bilgilerin tutar

B) SALONUN OLUŞTURULMASI:

- 446. Satırda salonu oluşturur. Salon, 20 elemanlı bir listenin içine “-----...-----” şeklinde 20 tane çizgiden oluşan bir String ataması ile elemanları String olan bir liste şeklinde oluşturulur. Her bir String salonun bir sırasını, her bir çizgi ise salonun bir koltuğunu temsil eder.
- Salonun diğer genel bilgileri olan kategorilere göre kalan bilet sayısı (**kalanBilet[]**) ve kategorilere göre elde edilen ciro (**toplamCiro[]**) tanımlanır. Tıpkı ücretler[]’de olduğu gibi 0.indiste 1.kategorinin bilgileri, 1.indiste 2.kategorinin bilgileri tutulur.

C) ANA MENU:

- While döngüsü ile kullanıcı çıkışa basmadıkça (girilen=”0”) programın sürdürülmesi sağlanır.
- Girilen input değerine göre **rezervasyon()**, **salonYazdir()**, **yeniEtkinlik()**, **ciroHesaplama()** fonksiyonlarından biri çağrılır.
- Tanımlanan giriş değerlerinden farklı bir değer girildiğinde hata mesajı gösterilerek tekrar seçim yapılması istenir. Bu geçerli bir değer girilene kadar devam eder.

Ana menüde çağrılan fonksiyonlar ve işlevleri:

- **salonYazdir(salon):** parametre olarak gönderilen salonun ekrana güncel görüntüsünü bastırır.
- **yeniEtkinlik(salon):** salonda yapılmış değişiklikleri salonu yeniden oluşturarak (452.satırdaki gibi) rezerve işlemlerini sıfırlar. **toplamCiro** ve **kalanBilet** bilgilerini de başlangıç değerlerine döndürür.
- **ciroHesaplama():** **toplamCiro** ile kategorilere göre ve toplam ciro bilgisini ekrana bastırır
- **rezervasyon(salon):**
 - while döngüsü ile kullanıcıdan kategori (**seçimKategori**) ve bilet (**seçimBiletAdet**) bilgisi alır. Kategori sayısı 1-4 arasında olana kadar ve bilet sayısı **biletMax** değişkenine uygun girilene kadar sormaya devam eder.
 - **seçimKategori** değerine göre ilgili kategoride istenilen bilet sayısı için yeterince boş koltuk var mı kontrol eder. Eğer yoksa ekrana yetersiz koltuk mesajı bastırır. Eğer koltuk sayısı varsa önce ilgili kategorinin rezervasyon fonksiyonunu çağırır. Daha sonra **ucretHesapla** fonksiyonunu çağırır.

D) REZERVASYON İŞLEMİ

Tüm rezervasyon fonksiyonları 3 aşamada çalışır.

1. En son bilet rezerve edilen koltuğun bulunması
2. Bu koltuğa rezervasyon işlemine başlanacak koltuğun değerlerinin atanması
3. Rezervasyon işlemi

Çalışma şekilleri aynı olduğu için kategori 1 ile kategori 3 aynı anda açıklanacaktır:

a. Kategori 1 ile Kategori 3: `rezerveKategori_1()` ve `rezerveKategori_3()`

--Aşama 1: en son rezerve edilen koltuğun indisi bulunur

```
def rezerveKategori_1(salon, biletNo): #1. kategori için rezervasyon işlemini yapar
    counter = 10
    # ---EN SON HANGİ KOLTUK REZERVE EDİLMİŞ ONU BULMAK İÇİN---
    for i in salon[9::-1]: #son rezervayı bulmak için en son sırada sağdan sola doğru x değeri arar, bulamazsa üst sıraya geçer
        counter = counter - 1 #counter sıra indisini tutar
        a = i.rfind("x")
        if a>=15: #2. kategori sağ taraftan x algıladı
            a=salon[counter][5:15].rfind("x")
            if a<0: #demek ki bu sırada 1. kategori boş,aramaya üst sıradan devam edilir
                continue
            else: #1. kategori bu sırada son koltuk bulundu,döngüden çıkılır
                a=a+5
                break
        elif a<=4: #2. kategori sol taraftan algıladı, 1. kategori bu sırada boş, aramaya üst sıradan devam edilir
            continue
        elif a >=5 and a<=14: #2. kategori sağ tarafta x algılanmadı, 1. kategori için aranan son koltuk bulundu, döngüden çıkıldı
            break
```

- Salonun o kategorideki son sırasından başlanır. Her sıra için sağdan sola doğru bakılarak “x” değeri aranır. Aranılan değer (a) bulunmazsa bir üst sıraya geçilir. Koltuğun indisi a değişkeni ile tutulurken sıra numarası da counter değişkeni ile takip edilir. Rezerve edilecek kategorilerin ortada olduğundan dolayı sağ ve soldaki kategorilerin “x” değerlerinin aranılan değere karışmaması için if-else blokları ile gerekli kontroller yapılır. a için kontrol edilen olasılıklar:
- (a>=15) ise 2./4. Kategori sağ kısımdan “x” algılandı:
 - O sıra için orta kategoriye bakılır. Eğer “x” değeri bulunamazsa bir üst sıraya geçilir. Eğer bulunursa son koltuk bulunduğundan döngüden çıkılır.
- (a<=4) ise 2./4. Kategori sol kısımdan “x” algılandı:
 - Sol kısımdan “x” bulunursa bu orta kısımda o sıra için hiç rezerve yapılmamış demektir, üst sıraya geçilir.
- (5<=a<=14) ise 1./3. Kategoriden “x” algılandı:
 - Aranılan koltuk bulunduğundan döngüden çıkılır.

--Aşama 2: a ‘nın değerine göre biletBaslangic ve sıraBaslangic değerleri atanır.

```
77 # ---EN SON REZERVE EDİLEN KOLTUĞA GÖRE REZERVASYON İŞLEMİNE BAŞLANACAK SIRA VE KOLTUK İNDİSLERİNİN BELİRLENMESİ---
78 if (a < 0): # ilk rezervasyon işlemi
79     biletBaslangic = 5
80     sıraBaslangic = 0
81 elif (a == 14): # en son sıranın sonuncu koltuğu rezerve edildiyse bilet başlangıç değerine bir sonraki sıranın ilk koltuğu atılır
82     biletBaslangic = 5
83     sıraBaslangic = counter + 1
84 else:
85     biletBaslangic = a + 1
86     sıraBaslangic = counter
```

sıraBaslangic: Rezerve edilmeye başlanacak yerin sıra indisini tutar.

biletBaslangic: Rezerve edilmeye başlanacak yerin koltuk indisini tutar.

- a<0 ise kategorinin ilk rezervasyon işlemi yapılacak demektir. Başlangıç değerleri ilk sıra ilk koltuk olarak atanır.
- a==14 ise sıra sonundaki koltuk rezerve edilmiş demektir. Başlangıç değerleri bir sonraki sıranın ilk koltuğu olarak atanır.
- a ile aynı sıradaki bir sonraki koltuk başlangıç koltuğu olarak belirlenir.

--Aşama 3: Rezerve İşlemi

Bilet rezerve işleminde oluşacak farklı durumlar için farklı if-else blokları oluşturulmuş, bilet ataması ise for döngüsü ile yapılmıştır. Kategori 1 ve 3 için iki farklı durum vardır: bilet rezerve işlemi başlanan sırada bitecek ya da başlanan sıra dolduktan sonra alt sıraya geçilecektir.

```
88 # ----BİLET REZERVE KISMI----
89 if (biletBaslangic + biletNo > 15): # rezerve edilecek koltukların bir kısmı alt satıra geçecek
90     counter = 0
91     for i in range(0, (15 - biletBaslangic)):
92         counter = counter + 1 # counter kaç bilet rezerve edildiğini tutuyor
93         salon[siraBaslangic] = salon[siraBaslangic][:biletBaslangic + i] + "x" + salon[siraBaslangic][biletBaslangic + 1 + i:]
94         print(f"({siraBaslangic+1},{biletBaslangic+i+1})",end="")
95     siraBaslangic = siraBaslangic + 1 # alt satıra geçildi
96     biletBaslangic = 5
97     for i in range(0, (biletNo - counter)): # biletNo-counter=kaç tane daha rezerve edilmesi gereken bilet var
98         salon[siraBaslangic] = salon[siraBaslangic][:biletBaslangic + i] + "x" + salon[siraBaslangic][biletBaslangic + 1 + i:]
99         print(f"({siraBaslangic+1},{biletBaslangic+i+1})",end="")
100 else: #aynı sıra içinde kalacak
101     for i in range(biletNo):
102         salon[siraBaslangic] = salon[siraBaslangic][:biletBaslangic + i] + "x" + salon[siraBaslangic][biletBaslangic + 1 + i:]
103         print(f"({siraBaslangic+1},{biletBaslangic+i+1})",end="")
```

-Aynı sırada kalınacaksa bilet sayısı (**biletNo**) kadar for döngüsü çalıştırılır, her döngüde **biletbaşlangıç+i** ile bir sonraki koltuğa geçilir. Döngüden çıkmadan önce rezerve edilen koltuğun bilgileri ekrana bastırılır.

-Alt sıraya geçilecekse önce yukarıdaki gibi ilk sıra for döngüsü doldurulur. Bu döngünün içinde daha sonra kaç bilet sayısı kaldığını bulabilmek için **counter** değişkeni artırılır. Daha sonra **siraBaslangic** bir artırılarak sonraki sıraya geçilir, **biletBaslangic** değişkenine ise sıranın ilk koltuğu olan 5 atılır. Kalan bilet sayısı **biletNo-counter** ile bulunarak ikinci bir for döngüsü açılır ve aşağıdaki sıra da doldurulur.

Çalışma şekilleri aynı olduğu için kategori 2 ile kategori 4 aynı anda açıklanacaktır:

a. Kategori 2 ile Kategori 4: **rezerveKategori_2()** ve **rezerveKategori_4()**

--Aşama 1: en son rezerve edilen koltuğun indisi bulunur

```
105 def rezerveKategori_2(salon, biletNo): #2. kategori için rezervasyon işlemini yapar
106     counter = 10
107     # ----EN SON HANGİ KOLTUK REZERVE EDİLMİŞ ONU BULMAK İÇİN----
108     for i in salon[9::-1]: #son rezerveyi bulmak için en son sırada sağdan sola doğru x değeri arar, bulamazsa üst sıraya geçer
109         counter = counter - 1 #counter sıra indisini tutar
110         a = i.rfind("x")
111         if a >= 15: # sağ tarafta kalınan yeri buldu, döngüden çıkılır
112             break
113         elif a >= 5: #1. kategorinin x'ini tespit etti,2. kategori sol taraf için bakması lazım
114             a=salon[counter][0:5].find("x")
115             if counter!=0 and a<0: #bu sırada hiç 2. kategoriden rezerve yapılmamış üst sıraya geçilir
116                 continue
117             elif counter!=0 and a<=4: #2. kategoride sol tarafta nerede kalındığını buldu, döngüden çıkılır
118                 break
119             elif counter==0 and a<0: #daha önce 2. kategoriden hiç rezerve yapılmamış->ilk rezerve işlemi
120                 break
121         elif a<=4 and a>=0: #sağ taraf boş solda rezerve yapılmış koltuk buldu, döngüden çıkılır
122             break
```

- Koltuğu bulmak kategori 1 ve 3 için gerçekleşen işlemlerin aynısı uygulanır. Burada farklı olan kontrol edilecek olasılıklardır. Kategori sağ ve sol taraf olarak bölündüğü için "x" değerinin algılanacağı 3 temel olasılık vardır:
- (**a**>=15) ise 2./4. Kategoride sağ kısımdan "x" algılandı:
 - Aranılan değer bulundu, döngüden çıkılır.
- (**a**>=5) ise 1./3. Kategoriden algılandı, aynı sırada 2. Kategori sol taraf için tekrar bakılır:
 - **a** negatif çıkarsa o sırada sol taraftan hiç rezerve yapılmamıştır, üst sıradan döngü devam eder.
 - **a** negatif çıkarsa ve counter==0 ise bu durumda en üst sıraya gelinmiştir ve şimdiye kadar bir "x" değeri bulunmamıştır. Yani 2. Kategori için ilk rezerve işlemi yapılacaktır.
 - **a** pozitif çıkarsa sol tarafta kalınan yer bulunmuştur, döngüden çıkılır.

- (4>=a>=0) ise 2./4. Kategori sol kısımdan “x” algılandı:
 - Aranan değer bulundu, döngüden çıkılır.

--Aşama 2: a 'nın değerine göre **biletBaslangic** ve **siraBaslangic** değerleri atanır.

```

123 # ----EN SON REZERVE EDİLEN KOLTUĞA GÖRE REZERVASYON İŞLEMİNE BAŞLANACAK SIRA VE KOLTUK İNDİSLERİNİN BELİRLENMESİ----
124 if a < 0: # ilk rezervasyon işlemi
125     biletBaslangic = 4
126     sirasBaslangic = 0
127 elif a == 19: # sıranın en son koltuğunda kaldıysa alt sıraya geçiş yapılır
128     biletBaslangic = 4
129     sirasBaslangic = counter + 1
130 elif a >= 15: # sağ tarafta kaldıysa
131     biletBaslangic = a + 1
132     sirasBaslangic = counter
133 else: # sol tarafta kaldıysa
134     a = salon[counter].find('x') # solda kalan yer bulunur
135     if a > 0: #demek ki solda hala bilet rezerve edilecek yer var
136         biletBaslangic = a - 1
137         sirasBaslangic = counter
138     else: #sol tamamen dolu sağ kısma geçiş sağlanır
139         biletBaslangic = 15
140         sirasBaslangic = counter

```

siraBaslangic: Sıra indisini tutar

biletBaslangic: Rezerve edilmeye başlanacak yerin koltuk indisini tutar.

- -a 'nın değerine göre nerede kalındığı belirlenir. Buna göre başlangıç değerleri yukarıdaki şekilde atanır.

--Aşama 3: Rezerve İşlemi

Bilet rezerve işleminde oluşacak farklı durumlar için farklı if-else blokları oluşturulmuş, bilet ataması ise for döngüsü ile yapılmıştır. Kategori 2 ve 4 için bilet sayısına (**biletNo**) ve başlangıç koltuk numarasına (**biletBaslangic**) göre yaşanabilecek durumların listesi aşağıdadır:

-Bilet rezervasyonu soldan başlayacak:

- Rezervasyon sağ tarafa geçmeden sadece sol tarafta yapılacak
- Rezervasyon sol taraf dolduktan sonra sağ tarafa geçecek:
 - Rezervasyon sağ tarafta bitecek
 - Rezervasyon sağ taraf dolacağı için sol tarafta bir sonraki satıra geçecek

-Bilet rezervasyonu sağdan başlayacak:

- Rezervasyon sadece sağ tarafta gerçekleşecek
- Rezervasyon sağ taraf dolduktan sonra sol tarafa geçecek
 - Rezervasyon sol tarafta bitecek
 - Rezervasyon sol taraf dolduktan sonra sağ tarafa geçecek

Bu durumlara göre girilen if-else bloklarında for döngüleri ile biletler rezerve edilir. Her sağ ve sol tarafa geçişte **biletBaslangic** ve **siraBaslangic** değerleri geçişi sağlamak için güncellenir. Geçiş esnasında kalan bilet sayısı **counter** değişkeni sayesinde hesaplanır. Her rezerve edilen koltuk for döngüsü içindeyken ekrana bastırılır.

E) ÜCRETİN HESAPLANMASI

`rezervasyon()` fonksiyonunun çağırdığı `rezerveKategori_1/2/3/4()` sonlandıktan sonra `ucretHesapla()` fonksiyonu çağrılır. Bu fonksiyona parametre olarak atılan kategori numarasına göre ilgili kategorinin indirim bilgisini tutan liste `i` isimli bir listeye atılır. `i` listesi ile indirim, net tutar, toplam tutar gibi bilgiler hesaplanır. En sonda `toplamCiro` bilgisini hesapladıklarına göre günceller.

```
353 def ucretHesapla(biletSayisi,kategoriNumarasi): #ücret,indirim,net tutar bilgilerini hesaplar
354     global ucretler
355     global liste1
356     global liste2
357     global liste3
358     global liste4
359     global toplamCiro
360     global toplamCiro_genel
361     i=[]
362     if kategoriNumarasi==1: #kategori numarasına göre indirim bilgisini alır
363         for x in range(0,3):
364             list=[]
365             for y in liste1[x]:
366                 if y=='M':
367                     y=5
368                 list.append(int(y))
369             i.append(list)
370 > elif kategoriNumarasi==2: #kategori numarasına göre indirim bilgisini alır...
378 > elif kategoriNumarasi==3: #kategori numarasına göre indirim bilgisini alır...
386 > else: #kategori numarasına göre indirim bilgisini alır...
394     print("\n--> Alınan bilet adeti:",biletSayisi)
395     tutar=biletSayisi*int(ucretler[kategoriNumarasi-1]) #tutarı hesaplar
396     print("--> Toplam tutar: ",tutar)
397 > if biletSayisi<int(i[0][1]): #bilet sayısına göre indirim miktarını ve net tutarı hesaplar
398     print("--> İndirim miktarı: %0")
399     print("--> Net tutar:",tutar)
400 > elif biletSayisi<=int(i[0][2]): #bilet sayısına göre indirim miktarını ve net tutarı hesapla...
405 > elif biletSayisi<=int(i[1][2]): #bilet sayısına göre indirim miktarını ve net tutarı hesapla...
410 > else: #bilet sayısına göre indirim miktarını ve net tutarı hesapla
411     indirim=int(i[2][3])
412     print("--> İndirim miktarı: %",indirim)
413     tutar=tutar-((10*tutar)/100)
414     print("--> Net tutar: ",tutar)
415     toplamCiro[kategoriNumarasi-1]+=tutar #tutarı ilgili kategorinin cirosuna ekler
```