

**MARMARA ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİLGİSAYAR PROGRAMLAMA 2**

**2022-2023 GÜZ YY. DÖNEM PROJESİ RAPORU**

**ÖĞRENCİ:**

**AD: Nuri Can**

**SOYAD: Birdemir**

**ÖĞR.NO: 170421843**

**DANIŞMAN:**

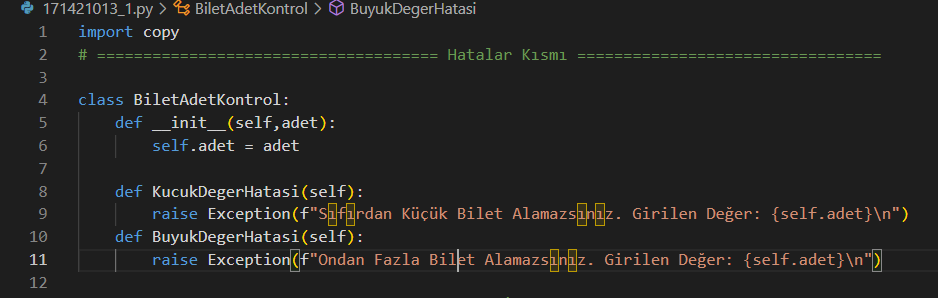
**Dr. Öğr. Üyesi Ayşe Berna ALTINEL**

**Araş. Gör. Merve PINAR**

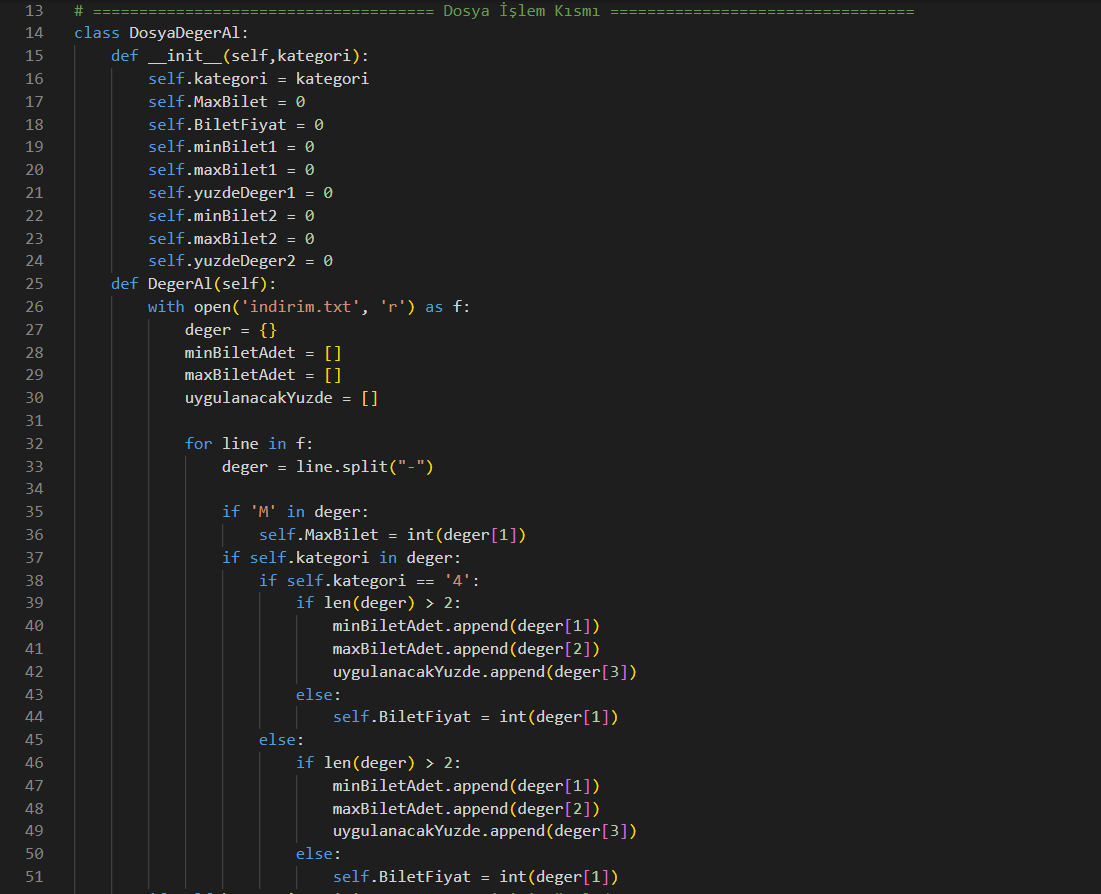
**Araş. Gör. Şeyda KARCI**

Aralık 2022

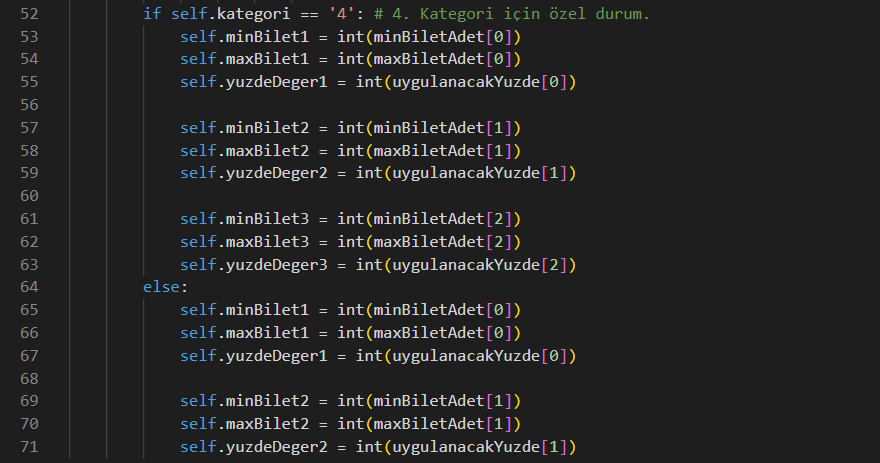
**Salon Rezervasyon Sistemi**

Proje hazırlanırken Visual Studio Code IDE’si kullanılmıştır. Projenin çalışma yapısında “Class” yapısı kullanılmıştır ve oluşturulmuş sınıf yapısından nesneler oluşturularak salon rezervasyon sistemi oluşturulmuştur.

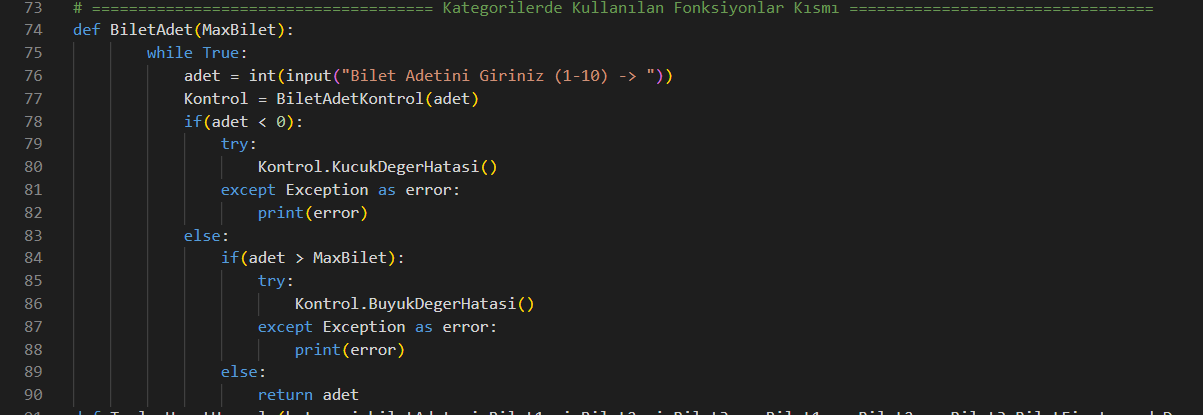
Resim Sistemde Bilet Alımında Oluşabilecek Hata Kontrol Sınıfı

Kod yapısı öncelikle kontroller ile başlamaktadır. Bu kontrollerden biri de hataların atandığı BiletAdetKontrol() Sınıf’tır. Bu sınıfta sistemde kullanıcıdan alınacak olan bilet adetinde (‘adet’ değişkeni) girilen bilet adetinin sıfırdan küçük (KucukDegerHatasi()) ve en büyük alınabilecek olan bilet adetinden fazla bir değer girilmesi (BuyukDegerHatasi()) durumda fırlatılan hata mesajları kontrol edilmektedir.

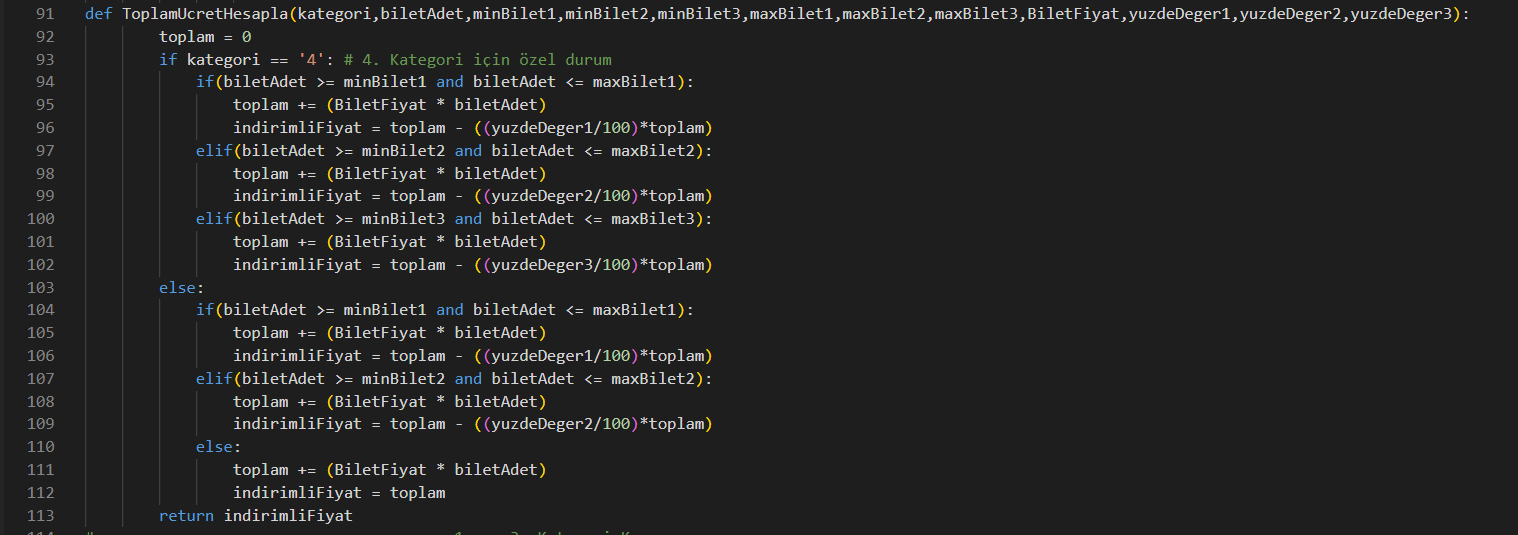
Resim İndirim.txt Dosyasından Kategorilere Özgü Değerleri Alan Sınıf

DosyaDegerAl() sınıfı ilk olarak dosyada bulunan değerleri bir değişkende tutmak için öncelikle bu değişkenler oluşturulmaktadır. DegerAl() metodu ile ‘indirim.txt’ dosyası okuma modunda açılarak ‘line’ adlı değişkene satır satır atama yapmaktadır. Sonra ‘deger’ adlı dictionary listesine dosyada tutulma şekli olan ‘-‘ karakterinden ayırarak ‘deger’ adlı listemize key – value (Anahtar kelime - değer) olarak atama yapılmaktadır. Atamalar bittiğinde en büyük ne kadar bilet alacağımızı tuttuğumuz ‘M’ adlı harfi ‘deger’ listesinde bulduğunda onun değeri olan value sayısını ‘MaxBilet’ değişkenine atılmaktadır. Devam eden kontrollerde ‘indirim.txt’ dosyasında kategorilerin hangi bilet sayısı aralığında yüzde kaç indirim uygulanacağını yazan değerleri dosyadan alırken oluşturduğumuz ‘minBiletAdet’,’maxBiletAdet’ ve ‘uygulanacakYuzde’ listelerine öncelikle uygulanacak indirimler için kaç adet olduğunu kontrol ederek alınmaktadır. Sebebi ise örnek olarak bu sistemde ‘4.’ Kategori için 3 adet indirim uygulama tarifesi atanmaktadır. Diğer durumlar için 2 adet indirim tarifesi uygulama atanmaktadır. Eğer indirim tarifesine ekleme yapılmak istenirse bu durumlar için program yine doğru çalışmaktadır. Bu durum kontrolü de yapıldığından sonra oluşturduğumuz listelere sıra ile aldığımız en küçük bilet adetini ‘minBiletAdet’ listesine atanır, buna benzer şekilde ilerleyen sıra ile diğer ‘maxBiletAdet’ ve ‘uygulanacakYuzde’ listelerine değerler eklenmektedir.

Resim ‘indirim.txt’ Dosyasından Alınan Değerlerin Değişkenlere Atanması İşlemleri

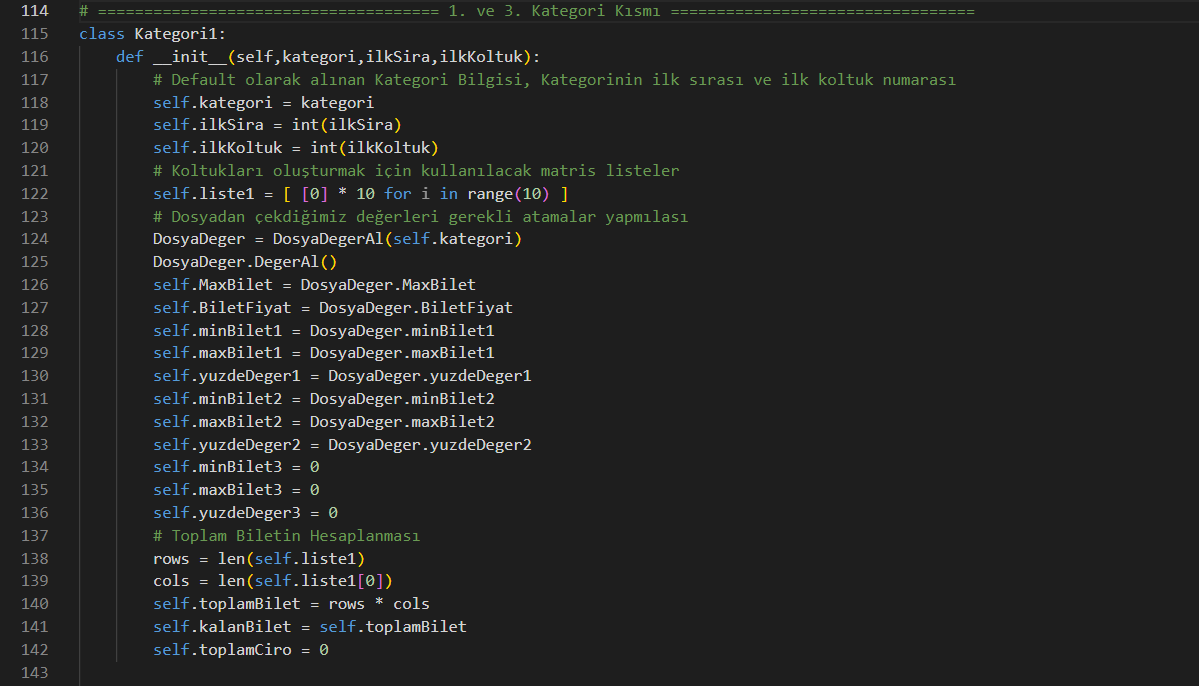
Resim 2 için anlattığımız olaylardan sonra listelerimize aldığımız değerlerin ‘4.’ Kategoriye özgü olan yapısı için ve diğer kategorilerin bilgisini atamak için oluşturduğumuz ‘minBilet1’, ’maxBilet1’, ‘yuzdeDeger1’ değişkenleri benzeri olan 2 ve 3 sonlu aynı ada sahip değişenlere sıra ile tutulan kategoriye özgü değerler atanır ve böylece sistemimizde kullanılacak olan kategorilere özgü indirim tarifeleri değişkenlerimize atanmış olmaktadır.

Resim Bilet Adetinin Alındığı ve Kontrol Edilen Fonksiyon

BiletAdet() metodu Resim 1’de anlattığımız kontrolleri içeren ve kullanıcıya kaç bilet almak istediğini soran metot olmaktadır. Burada kullanıcının verdiği cevabı sıfırdan küçük olma ve parametre olarak da aldığı ‘MaxBilet’ değişkeninden büyük değer girilmesi durumunda Hata mesajı verip Resim 1’de anlatılan ‘BiletAdetKontrol’ sınıfının nesnesini ‘Kontrol’ adlı değişkene oluşturan ve bu nesne ile hataları çağıran nesne olmaktadır. Bu kontroller doğru değer girilene kadar devam ederek doğru değer girilmesi ile de çağrıldığı yere return ile doğru değeri gönderen metot olmaktadır.

Resim Alınan Biletlerin Uygun İndirimler Yapılması ile Hesaplanan Toplam Ücret Sınıfı

ToplamUcretHesapla() metodu girilen bilet adetini uygun indirim tarifelerine girerek bilet adetinin bilet fiyatı ile çarpılması ve toplam çıkan bu ücretten uygun indirim tarifesine girmesi durumunda hesaplanan toplam ücretten uygun yüzde değeri üzerinden indirim yaparak ödenecek tutarı gönderen metot olmaktadır. Değişkenler için ‘kategori’ değişeni ile uygun kategorinin indirim tarifesi yapılası sağlanmaktadır. ‘biletAdet’ ile hangi bilet adet sayısı arasına giren indirim yüzdesi ‘yuzdeDeger’ de tutulup, ‘toplam’ adlı değişkende hesaplanan ücretten ‘yuzdeDeger’ ile indirim uygulanarak hesaplama yapılmaktadır.



Resim Kategori 1 ve 3 Oluşumu ve İşlemleri İçeren Sınıf

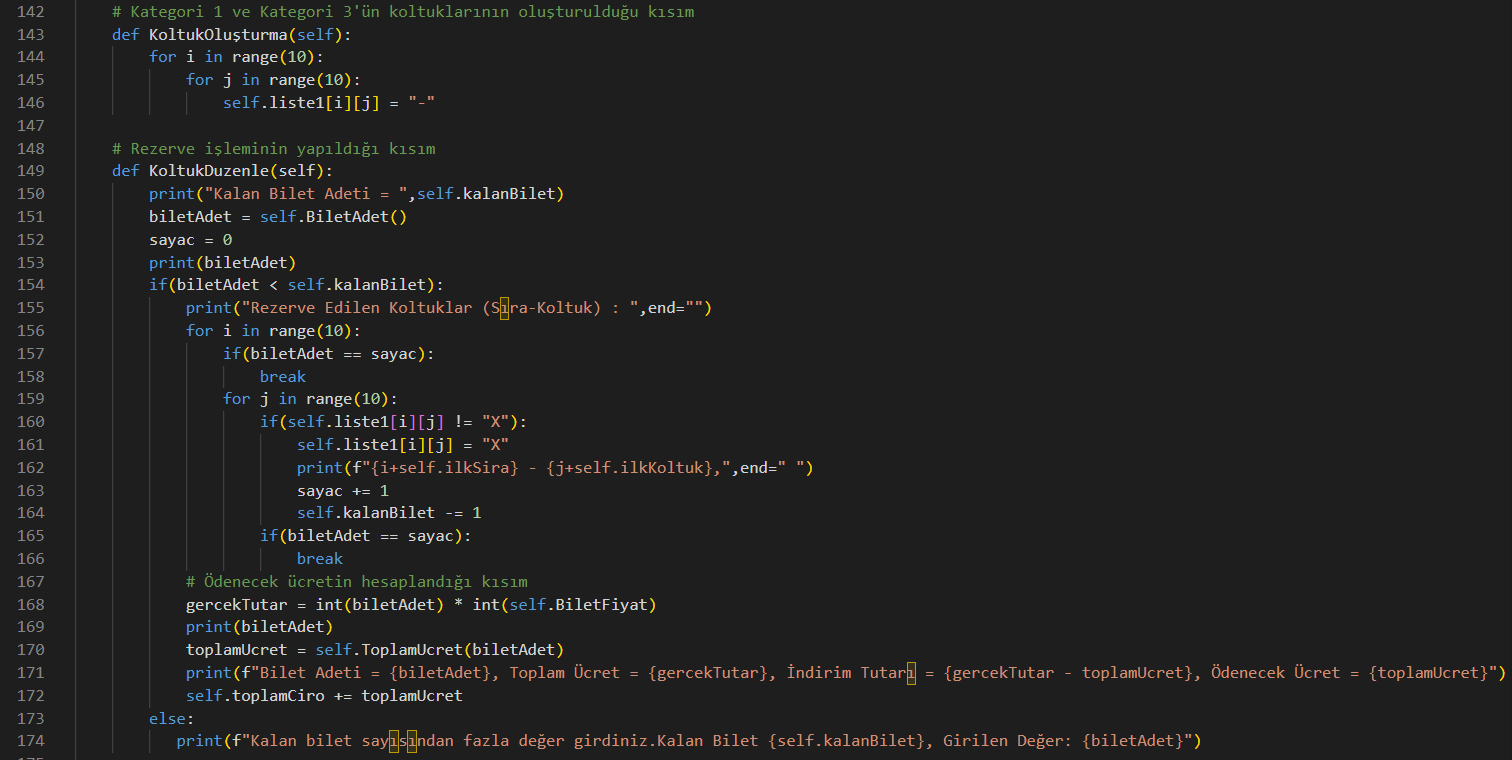
Kategori1 sınıfında sistemimizde 1. Kategori ve 3. Kategori oluşumu, rezervasyon işlemleri, koltukların doluluk oranı vb. işlemlerin yapılması işlemleri içermektedir.

Kategori1 sınıfının bir nesnesini oluşturmak için öncelikle ‘1’ veya ‘3’ kategori bilgisi girilip, bu kategorilerin ‘ilkSıra’ numarası ve ‘ilkKoltuk’ numarası gönderilir. Alınan bu bilgiler ile sonra 10x10’luk koltukların bulunacağı matris olan ‘liste1’ matris liste oluşturulur.

‘DosyaDeger’ adlı değişken Resim 2’ de anlatılan dosyadan değer çekme işlemlerinin yapıldığı DosyaDegerAl() sınıfının nesnesini alır. Nesne oluşumu ile oluşturulan değişkenlere değer alınan kategoriye uygun değerlerin atanmasını sağlayan DegerAl() metodu çağrılarak ilgili değerler atanmaktadır. Sonra Kategori1 sınıfında kullanılacak değerlere atama yapılmaktadır. 3. Bir indirim tarifesi 1. Kategori ve 3. Kategori için olmadığından şimdilik boş bırakılmıştır. Yeni indirim uygulanırsan gerekli değerler atanması yapılabilmektedir.

Değer ataması sonrası kategori için toplam koltuk sayısı hesaplanmaktadır. 10x10 bir kategori olduğu için 100 koltuk bulunmaktadır. ‘kalanBilet’ rezerve edilmemiş koltukların sayısını tutmaktadır.

‘toplamCiro’ oluşturulan kategorinin bilet satışından elde edilen geliri tutmaktadır.



Resim Kategori 1 ve 3 İşlemlerinin Yapıldığı Metotlar

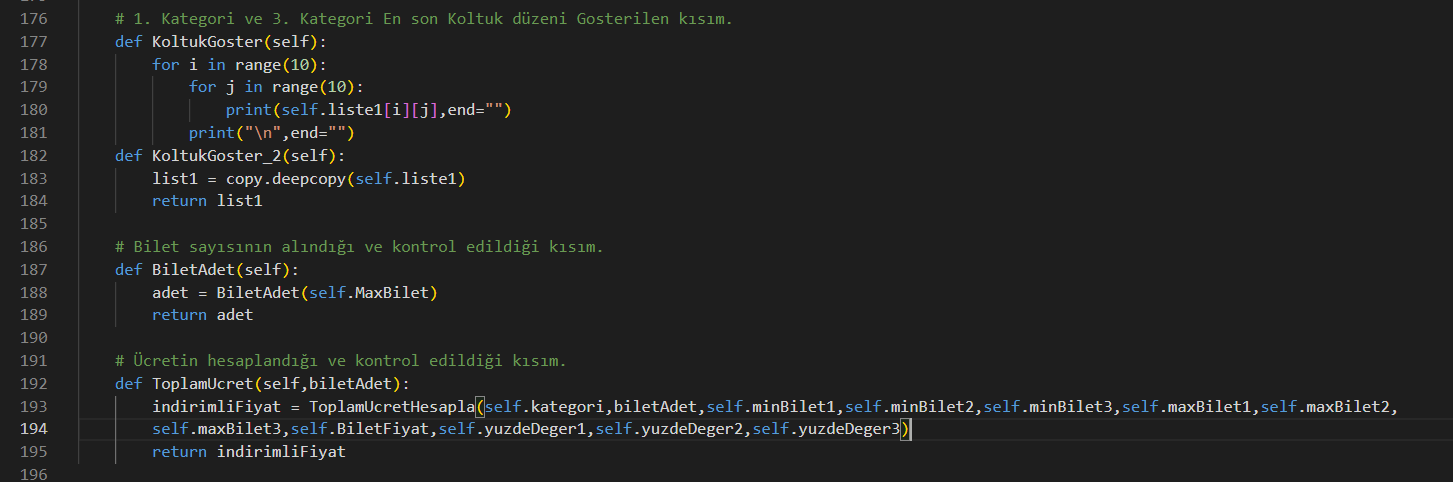
Resim 7 ‘Kategori1’ sınıfının işlemlerinin başlama yeri olmaktadır. Gerekli atamaların yapılmasını Resim 6’ da anlatılmaktadır.

KoltukOluşturma() metodu, ‘liste1’ adlı 10x10’luk matrisimize koltukların oluşumunu atamaktadır.

KoltukDuzenle() metodu, rezervasyon işlemlerini içermektedir. İlk olarak bize kategori için alınabilecek bilet sayısını yazdırmaktadır. Daha sonra kullanıcıdan bilet adeti istenmektedir. Bu işlemler Resim 4’de anlatılan metodun bulunduğu sınıf olan BiletAdet() sınıfından oluşturulan nesne olan ‘biletAdet’ e atanmaktadır. Gelen değer ‘kalanBilet’ten küçük ise rezervasyon işlemleri başlamaktadır.

‘sayac’ değişkenimiz bilet satışı yapıldıkça artarak ‘biletAdet’ nesnesine eşit olunca rezervasyon işlemlerini bitirir ve fiyat bilgisini yazdırmaya işlemlerine geçmektedir. Bu rezervasyon ‘liste1’ matrisimizdeki ‘-‘ olan boş koltukları bulunca ‘X’ ile değiştirip sayacı arttırarak yapmaktadır. Her bilet rezervasyonundan sonra ‘kalanBilet’ sayısı 1 azaltılmaktadır.

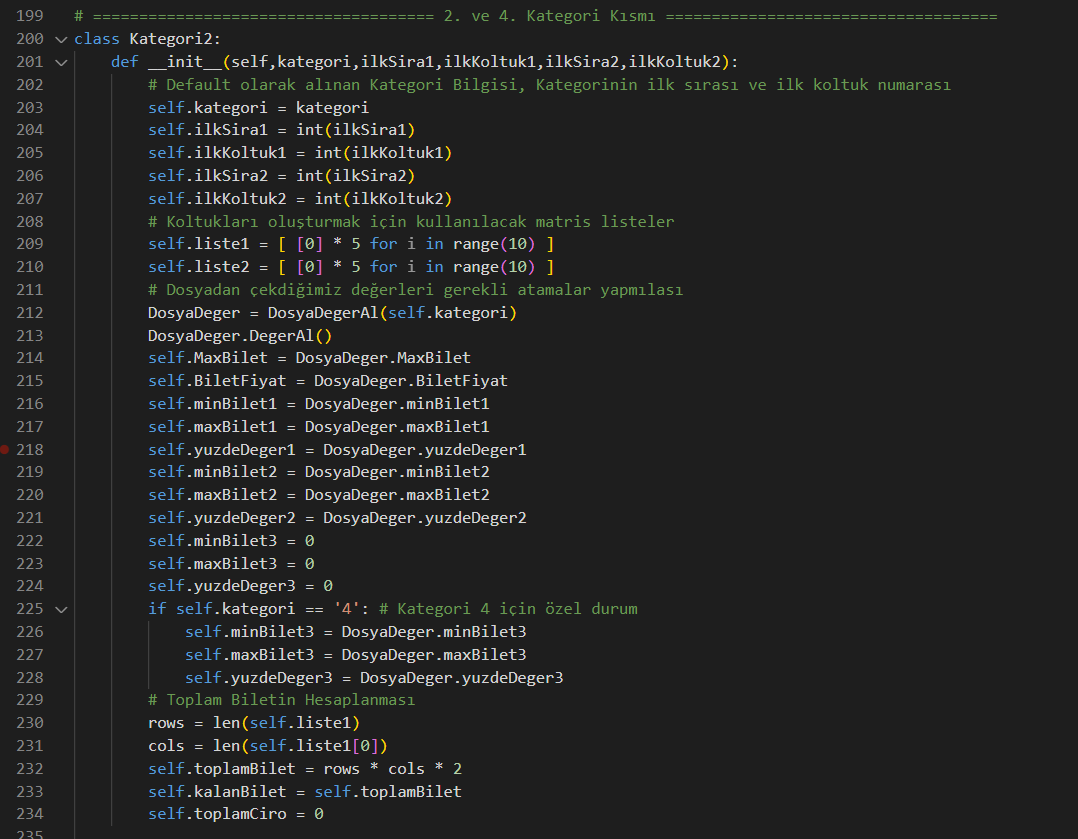
Ödenecek tutar hesaplamaları öncelikle indirimsiz fiyat hesaplanarak başlamaktadır. Resim 5’de de anlatılan indirim hesabının yapıldığı ‘ToplamUcretHesapla’ metodunun bulunduğu sınıf olan ‘ToplamUcret’ sınıfı ile indirimli fiyat bilgisi ‘toplamUcret’ değişkenine atanıp gerekli bilgi verme işlemi yapılmaktadır. ‘toplamCiro’ değişkenimize bilet satışındaki tutar eklenmiştir.

Eğer ‘kalanBilet’ rezerve edilmek istenen ‘biletAdet’inden az olursa gerekli uyarı mesajı verilmektedir.

Resim Kategori 1 ve 3 Koltuk Durumu Yazdırma ve İşlem Sınıfları

KoltukGoster() ve KoltukGoster\_2() metodu; KoltukGoster() metodu, kategori bilet satışı yapıldıktan sonra kategorinin son halini yazdıran metot iken KoltukGoster\_2() metodu salon genelini yazdırırken kullanılan metottur. KoltukGoster\_2() de ‘list1’ listemizi import ettiğimiz copy() sınıfından olan ‘deepcopy()’ işlemi ile nesne yapısından kurtulmuş yeni bir ‘list1’ listesi oluşturulmakta ve çağrıldığı yere return edilmektedir.

BiletAdet() sınıfı ve ToplamUcret() sınıfı kullanımı Resim 7’de anlatılmaktadır.



Resim Kategori 2 ve 4 Oluşumu ve İşlemleri İçeren Sınıf

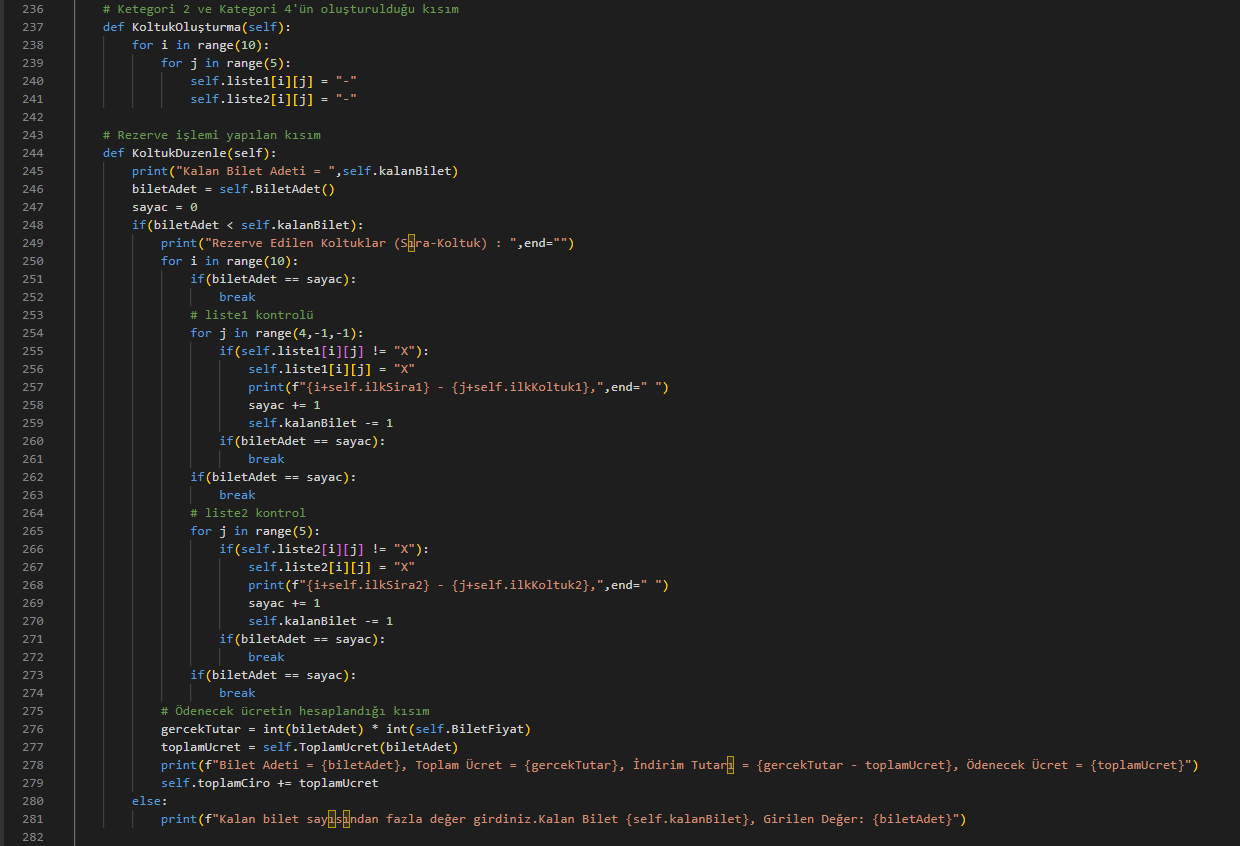
Kategori2 sınıfında sistemimizde 2. Kategori ve 4. Kategori oluşumu, rezervasyon işlemleri, koltukların doluluk oranı vb. işlemlerin yapılması işlemleri içermektedir.

Kategori2 sınıfının bir nesnesini oluşturmak için öncelikle ‘2’ veya ‘4’ kategori bilgisi girilip, bu kategorilerin ‘ilkSıra’ numarası ve ‘ilkKoltuk’ numarası gönderilir. Alınan bu bilgiler ile sonra 10x5’lik koltukların bulunacağı matris olan ‘liste1’ ve ‘liste2’ matris listeleri oluşturulur.

‘DosyaDeger’ adlı değişken Resim 2’ de anlatılan dosyadan değer çekme işlemlerinin yapıldığı DosyaDegerAl() sınıfının nesnesini alır. Nesne oluşumu ile oluşturulan değişkenlere değer alınan kategoriye uygun değerlerin atanmasını sağlayan DegerAl() metodu çağrılarak ilgili değerler atanmaktadır. Sonra Kategori2 sınıfında kullanılacak değerlere atama yapılmaktadır. Kategori 2 için iki adet indirim tarifesi bulunduğu için 3. İndirim tarifesi boş bırakılmaktadır. Eğer eklenmek istenirse program o tarifeyi uygulamaya hazır hale getirilmektedir. Kategori 4 için 3. Bir indirim tarifesi bulunduğu için bir ‘if’ yapısı ile o değerler alınmaktadır

Değer ataması sonrası kategori için toplam koltuk sayısı hesaplanmaktadır. 10x5 iki tane aynı numaradan kategori olduğu için 50-50 şeklinde koltuk bulunmaktadır. ‘kalanBilet’ rezerve edilmemiş koltukların sayısını tutmaktadır.

‘toplamCiro’ oluşturulan kategorinin bilet satışından elde edilen geliri tutmaktadır.



Resim Kategori 2 ve 4 İşlemlerinin Yapıldığı Metotlar

Resim 10, ‘Kategori2’ sınıfının işlemlerinin başlama yeri olmaktadır. Gerekli atamaların yapılmasını Resim 9’ da anlatılmaktadır.

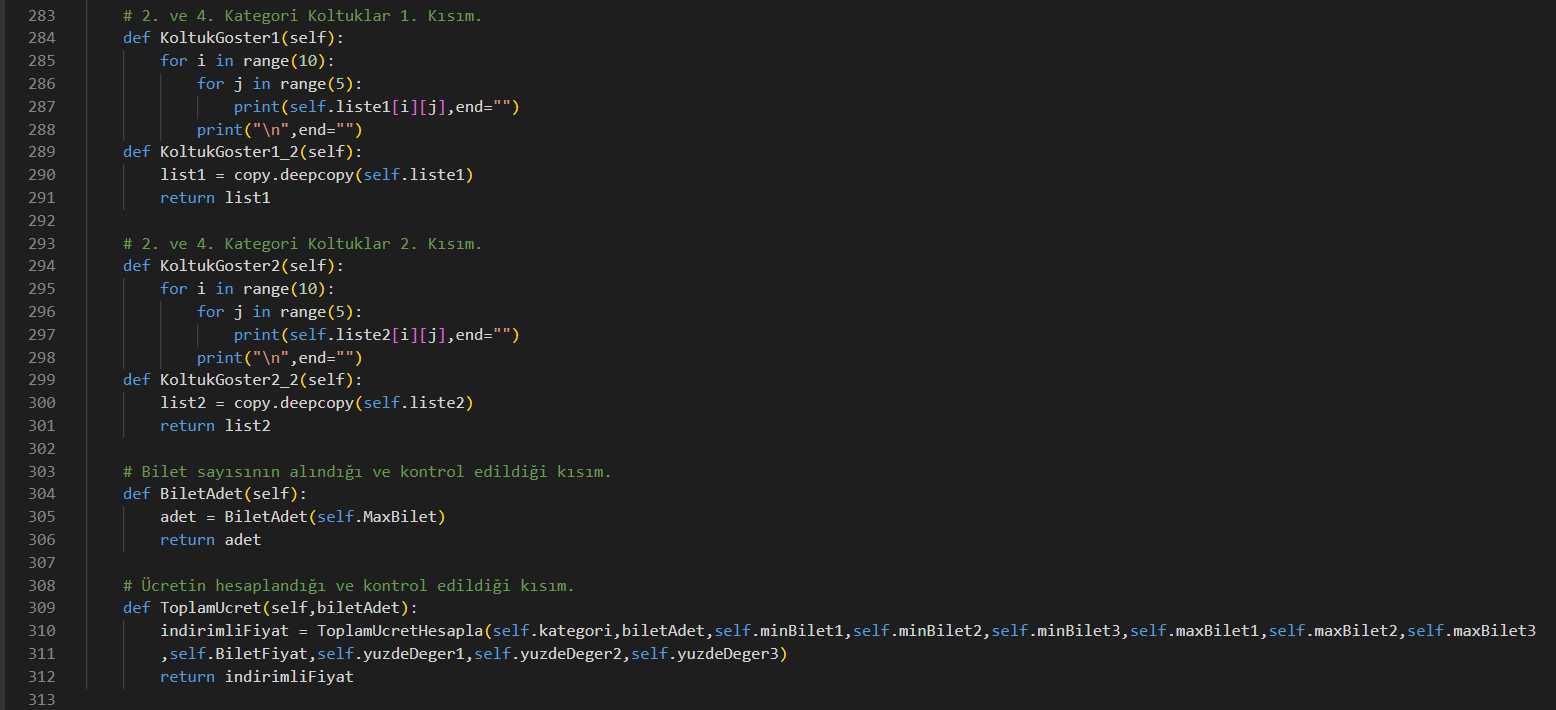
KoltukOluşturma() metodu, ‘liste1’ ve ‘liste2’ adlı 10x5’lik matrislere koltukların oluşumu atanmaktadır.

KoltukDuzenle() metodu, rezervasyon işlemlerini içermektedir. İlk olarak bize kategori için alınabilecek bilet sayısını yazdırmaktadır. Daha sonra kullanıcıdan bilet adeti istenmektedir. Bu işlemler Resim 4’de anlatılan metodun bulunduğu sınıf olan BiletAdet() sınıfından oluşturulan nesne olan ‘biletAdet’ e atanmaktadır. Gelen değer ‘kalanBilet’ten küçük ise rezervasyon işlemleri başlamaktadır.

‘sayac’ değişkenimiz bilet satışı yapıldıkça artarak ‘biletAdet’ nesnesine eşit olunca rezervasyon işlemlerini bitirir ve fiyat bilgisini yazdırma işlemlerine geçmektedir. Bu rezervasyon ‘liste1’ matrisimizdeki ‘-‘ olan boş koltukları bulunca ‘X’ ile değiştirip sayacı arttırarak yapmaktadır. Eğer bilet adeti 5’den büyük veya ‘list1’ de ilk indisi olan boş koltuk sayısından fazla girilirse ‘list1” listesinden çıkış yaparak ‘list2’ listesine gider ve oradan rezerve işlemlerini devam ettirerek istenen satış yapıldıktan sonra çıkış işlemi yapmaktadır. Her bilet rezervasyonundan sonra ‘kalanBilet’ sayısı 1 azaltılmaktadır.

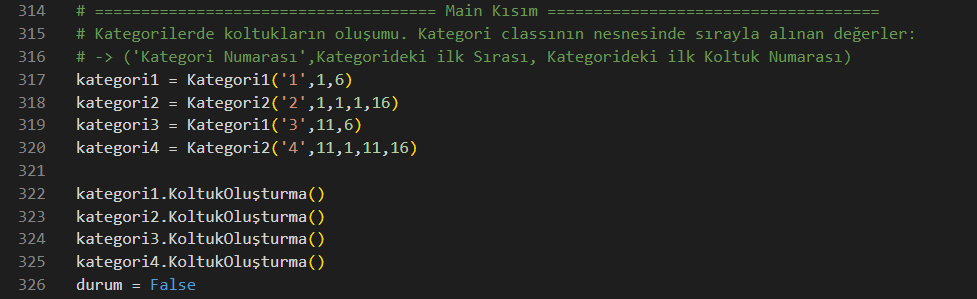
Ödenecek tutar hesaplamaları öncelikle indirimsiz fiyat hesaplanarak başlamaktadır. Resim 5’de de anlatılan indirim hesabının yapıldığı ‘ToplamUcretHesapla’ metodunun bulunduğu sınıf olan ‘ToplamUcret’ sınıfı ile indirimli fiyat bilgisi ‘toplamUcret’ değişkenine atanıp gerekli bilgi verme işlemi yapılmaktadır. ‘toplamCiro’ değişkenimize bilet satışındaki tutar eklenmiştir.

‘kalanBilet’ sayısı rezervasyon yapılmak istenen bilet sayısından küçük olursa gerekli mesaj verilmektedir.



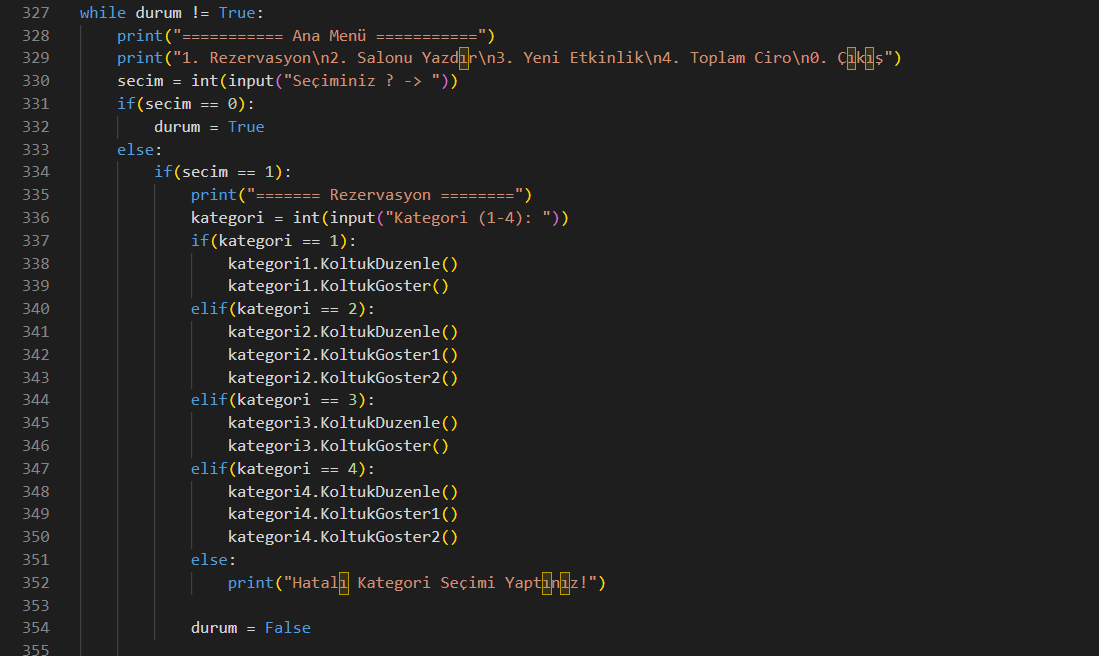
Resim Kategori 2 ve 4 Koltuk Durumu Yazdırma ve İşlem Sınıfları

Resim 11, Resim 8’de 1. Ve 3. Kategori için anlatılan durumundaki işlemleri 2 ve 4 için yapmaktadır.



Resim Sistemin Oluşturulduğu Main Kısmı

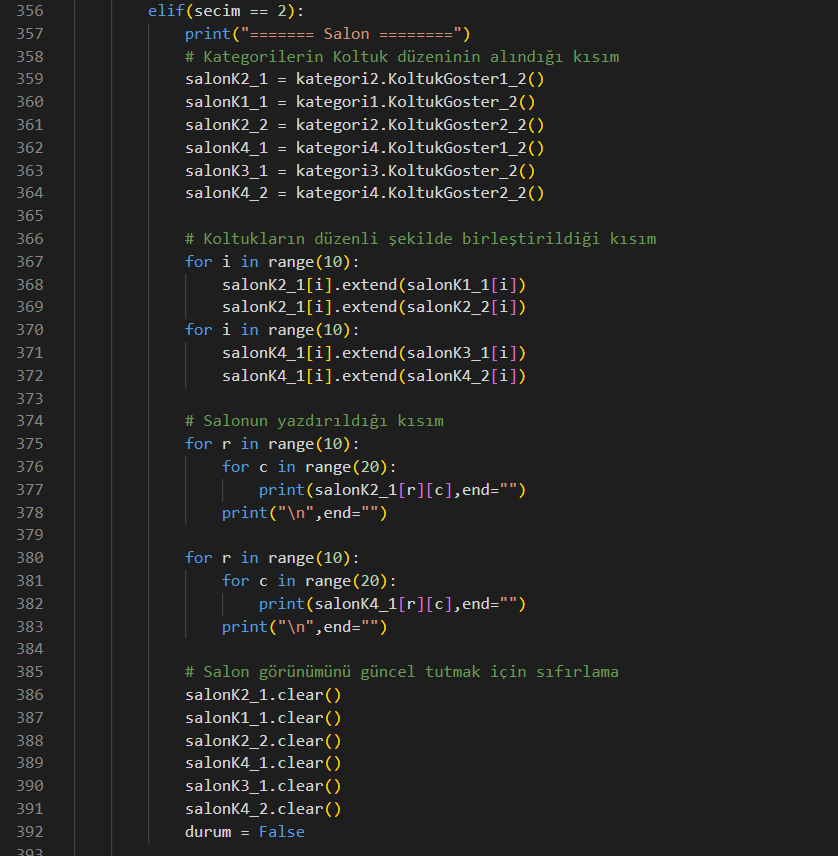
Main kısmında ilk olarak 4 kategorinin de nesnesi oluşturulmaktadır. Resim 6 ve Resim 9 da verilen parametrelerin sebebi açıklanmaktadır. Daha sonra bu 4 kategorinin de koltuklarını oluşturan metodu çağrılmaktadır. ‘durum’ değişkeni programın devamlılığını sağlan değişkendir.



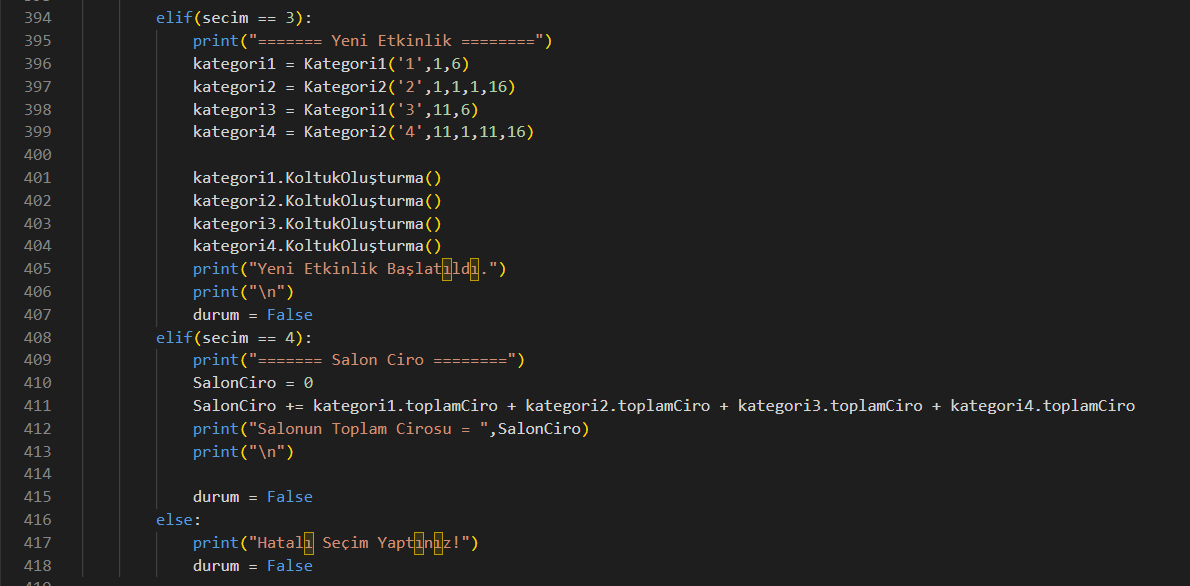
Resim Main Kısım İşlemleri

Mainde yapılacak olan Rezervasyon, Salon Yazdır, Yeni Etkinlik, Toplam Ciro ve Çıkış işlemlerinden hangi seçimi yapacağı kullanıcıya sorulmaktadır.

Rezervasyon işlemi, ‘1’ seçimi yapıldığında girilen kısımdır. Kategori seçimi yapılması istenerek ilgili kategorinin işlemleri başlatılmaktadır. KoltukDuzenle() metodu ile bilet satışı yapılmaktadır. ’KoltukGoster()’ ile seçim sonrası kategorinin durumu görüntülenmektedir.

 Seçim ‘2’ sonrası 4 kategorinin koltuklarının son durumu olan ‘KoltukGoster()’ metotları değişkenlere atanmaktadır. Liste olan bu değişkenler ‘extend’ metodu ile gerekli sıralama yapılarak salonun genel yapısı oluşturulmaktadır. Daha sonra ‘for’ döngüleri ile salon yazdırılmaktadır. Salonun her rezervasyondan sonra güncel kalması için değişkenlere atanan değerler temizlenmektedir.

Resim Salon Durumu Yazdırma



Resim Yeni Etkinlik ve Salon Ciro

Seçim ‘3’ ile bütün salon tekrar ‘KoltukOluştur()’ metodu ile oluşturulmaktadır. Seçim ‘4’ ile kategorilerde her bilet satışından sonra toplam satış geliri tutulan ‘toplamCiro’ değişkeni ile her kategorinin cirosu ‘SalonCiro’ değişkeninde toplanarak yazdırılmaktadır.