



T.C

KOCaeli SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ LİSANSÜSTÜ
EęİTİM ENSTİTÜSÜ
YAZILIM MÜHENDİSLİęİ PROGRAMI

ÖDEV KONUSU

YZM217-BLM217 Programlama Lab 1 - Ödev 2

Hazırlayanlar

ÖęRENCİ ADI: SERHAT ARSLANER

ÖęRENCİ NUMARASI: 220502043

GITHUB: <https://github.com/serhatarslaner>

ÖęRENCİ ADI: EMİR DURSUN

ÖęRENCİ NUMARASI: 220502001

GITHUB: <https://github.com/emirdrs>

DERS SORUMLUSU

Prof. Dr. H. Tarık DURU

TARİH 12.12.2023

İÇİNDEKİLER

ÖZET.....	2
GİRİŞ.....	2
KAYNAKÇA.....	10

ÖZET

Ödevimizi yaparken kütüphane , fonksiyon,sınıf,koşul ve döngü ifadeleri kullandık ödevin genel yapısında yapılması istenilen görevleri yerine getirmeye çaba gösterdik.

GİRİŞ

```
1 import time
2 import threading
3 from termcolor import *
```

Liman otomasyonunu gerçekleştirirken 3 tane kütüphane kullandık;

Time kütüphanesini,program çıktılarının verirken otomasyon akışının daha düzenli olması için sleep() fonksiyonuyla gecikme ekledik.

Threading kütüphanesini bir python uygulaması içerisinde bir veya birden fazla eş zamanlı kod çalışması için kullanılır.Biz bu kütüphaneyi tır yük indirme ve gemi yük bindirme işlemleri için kullandık. Termcolor kütüphanesini ise yük çıktılarına renk verebilmek,çıktıları kategorize etmek için kullandık.

```
5 # Dosya Okuma
6 gemiler=open("gemiler.csv","r",encoding="ISO-8859-9")
7 tirlar=open("olaylar.csv","r",encoding="ISO-8859-9")
```

Bilgilere erişmek için gemiler.csv ve olaylar.csv dosyalarını okuma modunda ve "ISO-8859-9" formatında açtık ve ilgili değişkenlere atama yaptık.

Ödev No: 2	Tarih 12.12.2023	
------------	------------------	--

```

10 # "gemiler.csv" düzenleme liste yapısı ve sözlük oluşturma.
11 gemiList=[]
12 for gemi in gemiler:
13     gemiVeri=gemi.split(",")
14     gemiVeriList=[i.strip("\n") for i in gemiVeri]
15     gemiList.append(gemiVeriList)
16     gemiList.pop(0)
17
18     gemiDict=dict()
19
20     for gemidictt in gemiList:
21
22         gemiDict[gemidictt[1]]=[gemidictt[0],gemidictt[2],gemidictt[3]]
23

```

Gemiler değişkenine atadığımız dosyayı for döngüsü içerisinde verilerini gezerek veriler üzerinden virgülle ayırdık (13.satır).Newline(\n)'ı silmek için strip() fonksiyonunu kullandık ve bu fonksiyon içerisinde verileri for döngüsü ile döndürdük,böylece newline kalmamış oldu (14.satır).Newline'ı sildiğimiz her elemanı yukarıda tanımladığımız gemiList adlı listeye ekledik,daha sonra döngü bittiğinde tüm verileri listeye eklemiş olduk,ilk satırdaki geliş_zamanı,gemi_adı gibi verileri gruplandıran satırı pop() fonksiyonu ile listeden çıkarttık(16.satır).Sonrasında elimizdeki verileri bizden istendiği gibi bir sözlük yapısına dönüştürdük.

```

24 # Gemi Verileri İçin Class
25 class Gemi():
26     def __init__(self,gemiAdı):
27
28         gemiAdıVeri=gemiDict[gemiAdı]
29
30         self.gemiAdı=gemiAdı
31         self.zaman=int(gemiAdıVeri[0])
32         self.kapasite=int(gemiAdıVeri[1])
33         self.ulke=gemiAdıVeri[2]

```

Gemiler.csv dosyası içerisindeki verilere erişebilmek için sınıf yazdık.Bu sınıfın tek parametresi var çünkü gemi adını girdiğimiz zaman diğer verilere de sözlük yapısı sayesinde erişim sağlayabiliyoruz.

```

37 # Plaka Sıralama
38 def plaka_sıralama(x):
39     return x[9:]

```

Tırların plakasını sıralamak için fonksiyon yazdık.Bu fonksiyon 41_kostu_ sonrasındaki 3 sayıyı ayırmamızı sağlıyor.Daha sonra ayrılmış plakaları sort() fonksiyonu sayesinde sıralayabiliyoruz.

Ödev No: 2	Tarih 12.12.2023	
------------	------------------	--

```

46 # "olaylar.csv" düzenleme liste yapısı ve sözlük oluşturma.
47 for tır in tırlar:
48     anlaktır=tır.split(",")
49     anlaktırlist=[i.strip("\n") for i in anlaktır]
50     olaylarverisi.append(anlaktırlist)
51
52 olaylarverisi.pop(0)
53 olaylardict=dict()
54
55 for olaylar in olaylarverisi:
56     if olaylar[3]=="0":
57         olaylardict[(int(olaylar[0]),olaylar[1])]=[olaylar[2],30000,olaylar[5],olaylar[6]]
58     else:
59         olaylardict[(int(olaylar[0]),olaylar[1])]=[olaylar[2],20000,olaylar[5],olaylar[6]]
60

```

Tırlar değişkenine atadığımız olaylar.csv dosyasındaki verileri for döngüsü içerisinde satır satır geziyoruz,ardından verileri split fonksiyonu sayesinde virgüllerden ayırıp satırları bir liste elemanı haline getiriyoruz.Sonrasında satır sonundaki newlineları silmek için her elemanı strip() fonksiyonu ile newlineları kaldırıyoruz.Sonrasında bu listeyi olaylarverisi isimli listemize ekliyoruz. ilk satırdaki geliş_zamanı,tır_plakası gibi verileri gruplandıran satırı pop() fonksiyonu ile listeden çıkarttık(52.satır).Sonrasında elimizdeki verileri bizden istendiği gibi bir sözlük yapısına dönüştürdük,20 tonluk veya 30 tonluk yüklerden biri 0 ise diğerinin maliyetini esas alıyor.

```

63 # Tır Verileri İçin Class
64 class Tır():
65     def __init__(self,zaman,plaka):
66         self.zaman=int(zaman)
67         self.plaka=plaka
68         veri=olaylardict[(zaman,plaka)]
69         self.ulke=veri[0]
70         self.kapasite=int(veri[1])
71         self.yük_miktari=int(veri[2])
72         self.maliyet=int(veri[3])
73
74 olaylarKeys = olaylardict.keys()
75 gemilerKeys = gemiDict.keys()
76

```

Olaylar.csv dosyası içerisindeki verilere erişebilmek için bir sınıf yazdık.Bu sınıfın iki parametresi var çünkü tır plakası ve zamanını girdiğimiz zaman diğer verilere de sözlük yapısı sayesinde erişim sağlayabiliyoruz.74 ve 75.satırda ise olaylar ve gemiler sözlük yapısındaki bütün verilere erişim sağlıyoruz.

Ödev No: 2	Tarih 12.12.2023	
------------	------------------	--

```

77 # İstif Alanı - Yük İndirme-Bindirme İşlemleri için Class
78 class istifAlani():
79     def __init__(self):
80         self.alan1 = 0
81         self.alan2 = 0
82         self.kapasite = 750
83         self.Neverland = 0
84         self.Mordor = 0
85         self.Lilliputa = 0
86         self.Oceania = 0

```

Gemiler ve tırlardan inecek veya yüklenecek malzemelerin kontrolünü ve liman otomasyonunun genel yapısını kontrol etmek için istifAlani() sınıfını kullandık.

```

def tırBilgileri(self,zaman,plaka,ulke,yukMiktari,maliyet):
    self.zaman = zaman
    self.plaka = plaka
    self.ulke = ulke
    self.yukMiktari = yukMiktari
    self.maliyet = maliyet
    print(colored("{} Zamanında Gelen {} Plakalı Tır {} Ülkesine Gitmek Üzere {} Ton Yük İndirdi. Maliyet = {}".format(zaman,plaka,ulke,yukMiktari,maliyet),"magenta"))

def gemiBilgileri(self,zaman,gemiAdi,kapasite,gidecekUlke):
    self.zaman = zaman
    self.gemiAdi = gemiAdi
    self.kapasite = kapasite
    self.gidecekUlke = gidecekUlke
    print(colored("{} Adlı Gemi {} Kapasite ile Limana Yaklaşıyor !!!!".format(gemiAdi,kapasite),"magenta"))

```

Gemi bilgilerine ve tır bilgilerine erişebilmek için 2 adet fonksiyon yazdık.Bu fonksiyonlar liman otomasyonu gerçekleşirken limana gidecek tır ve gemiler hakkında bilgi sahibi olmamıza yarıyor.

Ödev No: 2	Tarih 12.12.2023	
------------	------------------	--


```

104 def yükIndirme(self,miktar,ülke):
105     self.ülke = ülke
106     self.miktar = miktar
107     toplamYuk = self.alan1 + self.alan2
108
109     if miktar + toplamYuk <= 1500:
110         if self.ülke == "Neverland":
111             self.Neverland += miktar
112             if self.alan1 <= 750:
113                 if self.alan1 + miktar > 750:
114                     print("Yük Alan1'e Sığmadığı İçin Alan2'ye Aktarılıyor")
115                     self.alan2 += miktar
116                 else:
117                     self.alan1 += miktar
118             else:
119                 print("İstif Alan1 - 1 Dolu @ İstif Alan1 - 2'ye Yükleniyor...")
120                 if self.alan2 <= 750:
121                     self.alan2 += miktar
122                 else:
123                     print("İstif Alan1 - 2 Dolu @@@")
124
125         if self.ülke == "Mordor":
126             self.Mordor += miktar
127             if self.alan1 <= 750:
128                 if self.alan1 + miktar > 750:
129                     print("Yük Alan1'e Sığmadığı İçin Alan2'ye Aktarılıyor")
130                     self.alan2 += miktar
131                 else:
132                     self.alan1 += miktar
133             else:
134                 print("İstif Alan1 - 1 Dolu @ İstif Alan1 - 2'ye Yükleniyor...")
135                 if self.alan2 <= 750:
136                     self.alan2 += miktar
137                 else:
138                     print("İstif Alan1 - 2 Dolu @@@")
139
140         if self.ülke == "Lilliputa":
141             self.Lilliputa += miktar
142             if self.alan1 <= 750:
143                 if self.alan1 + miktar > 750:
144                     print("Yük Alan1'e Sığmadığı İçin Alan2'ye Aktarılıyor")
145                     self.alan2 += miktar
146                 else:
147                     self.alan1 += miktar
148             else:
149                 print("İstif Alan1 - 1 Dolu @ İstif Alan1 - 2'ye Yükleniyor...")
150                 if self.alan2 <= 750:
151                     self.alan2 += miktar

```

istifAlanı sınıfı içerisinde yazdığımız yükIndirme fonksiyonu init fonksiyonu içerisindeki alan1,alan2,kapasite ve ülkelere göre depolama değişkenleri üzerinden işlem yapmaktadır.Kendisine parametre olarak girilen miktar ve ülke,yükIndirme fonksiyonunun çalışması için gerekli olan ana bileşenlerdir.toplamYuk yani alan1+alan2 1500'den küçük ise yük indirme işlemi gerçekleşecektir.yükIndirme fonksiyonuna girilen ülke parametresi hangi ülkeye ait ise,yük indirme işlemleri ona göre yapılacaktır.Eğer alan1'in kapasitesi 750'den küçük veya eşitse yüklenecek yük alan2'ye aktarılacaktır.Ve bu şekilde her ülke için gerçekleşir.Toplam kapasite 1500'den büyükse istif alanları doluluğu hakkında geribildirim verecektir.

Ödev No:2	Tarih 12.12.2023	
-----------	------------------	--

```

171     def yükYükleme(self,gemiKapasitesi,gemiUlkesi,gemiZaman,gemiAdı):
172         self.gemiAdı = gemiAdı
173         self.gemiZaman = gemiZaman
174         self.gemiKapasitesi = gemiKapasitesi
175         self.gemiUlkesi = gemiUlkesi
176         maxKapasite = gemiKapasitesi*(95/100)
177
178         if self.gemiUlkesi == "Neverland":
179             if self.Neverland > gemiKapasitesi:
180                 self.Neverland -= gemiKapasitesi
181                 self.alan1 -= gemiKapasitesi
182                 print(colored(" GEMİ KALKIYOR {} ÜLKESİNE GİDİYOR".format(gemiUlkesi),"green"))
183                 if self.alan2 > gemiKapasitesi:
184                     self.alan2 -= gemiKapasitesi
185
186         if self.gemiUlkesi == "Lilliputa":
187             if self.Lilliputa > gemiKapasitesi:
188                 self.Lilliputa -= gemiKapasitesi
189                 self.alan1 -= gemiKapasitesi
190                 print(colored(" GEMİ KALKIYOR {} ÜLKESİNE GİDİYOR".format(gemiUlkesi),"green"))
191                 if self.alan2 > gemiKapasitesi:
192                     self.alan2 -= gemiKapasitesi
193
194         if self.gemiUlkesi == "Oceania":
195             if self.Oceania > gemiKapasitesi:
196                 self.Oceania -= gemiKapasitesi
197                 self.alan1 -= gemiKapasitesi
198                 print(colored(" GEMİ KALKIYOR {} ÜLKESİNE GİDİYOR".format(gemiUlkesi),"green"))

```

istifAlanı sınıfı içerisinde bulunan yükYükleme fonksiyonu 4 parametre almaktadır.gemiKapasitesi,gemiUlkesi,gemiZaman ve gemiAdı.yükİndirme fonksiyonunda gerçekleştirilen alan1 alan2 ve ülkelere göre depolama işlemlerine göre kendisine parametre olarak verilen gemiUlkesi,init fonksiyonu içerisinde bulunan ülkelere göre depolama değişkenleri üzerinden değişkenleri,koşullarını sağlıyorsa alan1'den veya alan2'den yük bindirme işlemleri gerçekleştirilecektir.

```

211     def alanKontrol():
212         if self.alan1 == 0:
213             print("İSTİF ALANI 1 = BOŞ")
214         if self.alan2 == 0:
215             print("İSTİF ALANI 2 = BOŞ")
216         alanKontrol()

```

alanKontrol fonksiyonu alan1 ve alan2 içerisinde yük bulunmuyor ise boş olduğu hakkında geribildirim verecektir.

Ödev No: 2	Tarih 12.12.2023	
------------	------------------	--

```

217 limanSimulasyonu = istifAlani()
218
219 # Yük İndirme Fonksiyonu
220 def olaylar():
221     for i in olaylarKeys:
222         tırS = Tır(i[0],i[1])
223         limanSimulasyonu.yükIndirme(tırS.yük_miktari,tırS.ulke)
224         limanSimulasyonu.tırBilgileri(tırS.zaman,tırS.plaka,tırS.ulke,tırS.yük_miktari,tırS.maliyet)
225         print(colored("İSTİF ALANI = ", "Alan 1 = ",limanSimulasyonu.alan1,"Alan - 2 = ",limanSimulasyonu.alan2),"cyan"))
226         time.sleep(0.2)
227 # Yük Yükleme Fonksiyonu
228 def gemiler():
229     for j in gemilerKeys:
230         gemiS = Gemi(j)
231         limanSimulasyonu.gemiBilgileri(gemiS.zaman,gemiS.gemiAdi,gemiS.kapasite,gemiS.ulke)
232         limanSimulasyonu.yükYükleme(gemiS.kapasite,gemiS.ulke,gemiS.zaman,gemiS.gemiAdi)
233         time.sleep(1)
234
235 # Liman Otomasyonu Çalıştırma
236 thread1 = threading.Thread(target=olaylar)
237 thread2 = threading.Thread(target=gemiler)
238 thread1.start()
239 thread2.start()
240 thread1.join()
241 thread2.join()

```

Limn otomasyonunun gerekleřtirilmesi bu ařamada oluyor.Olaylar fonksiyonunda szlk yapısına atadıđımız olaylar dosyası ierisinde geiziyoruz.Tır sınıfının ierisine atamak zere 1. ve 2. parametreyi gnderiyoruz.Bu sayede tırlar hakkındaki btn verilere ulařabiliyoruz.Bu parametreleri yukarıda anlatılan ykİndirme,tırBilgileri fonksiyonlarına gnderiyoruz.Ve istenilen ıktıyı ekrana bastırmak iin print() fonksiyonunu kullanıyoruz.Gemiler fonksiyonunda ise szlk yapısına atadıđımız gemiler.csv dosyasının ierisinde for dngs ile geiziyoruz.Ardından gemiler sınıfında eriřebilmek iin geizdiđimiz bu verileri gemi sınıfının ierisine atıyoruz.Ve yukarıda belirtilen gemiBilgileri ve ykYkleme fonksiyonuna gerekli parametreleri gnderiyoruz.


```

# Tır ve Gemi Bilgileri Sorgulama
sececek = input("Gemi Sorgulamak İçin = G, Tır Sorgulamak İçin = T tuşlayınız: ").lower()
if sececek == "g":
    def gemiSorgulama():
        gemiInput = input("Sorgulamak İstedığınız Geminin Adını Giriniz: ")
        gemiBilgi = Gemi(gemiInput)
        print("Gemi Adı =",gemiBilgi.gemiAdı,"Gemi Kapasite =",gemiBilgi.kapasite,"Gemi Ülkesi =",gemiBilgi.ulke,"Gemi Zaman =",gemiBilgi.zaman)

    gemiSorgulama()

if sececek == "t":
    def tirSorgulama():
        tirInput = input("Sorgulamak İstedığınız Tırın Zamanını Giriniz: ")
        tirInput1 = input("Sorgulamak İstedığınız Tırın Plakasını Giriniz: ")
        tirBilgi = Tır(tirInput,tirInput1)
        print("Kapasite =",tirBilgi.kapasite,"Maliyet =",tirBilgi.maliyet,"Plaka =",tirBilgi.plaka,"ÜLKE =",tirBilgi.ulke,"Yük Miktarı =",tirBilgi.yukMiktar)

    tirSorgulama()

```

Liman otomasyonu bittiği zaman,tır ve gemi bilgilerini sorgulamak için gemiSorgulama ve tirSorgulama adında 2 tane fonksiyon yazdık.Kullanıcıdan alınan verilere göre gemi veya tır sınıfının içerisine parametre olarak gönderiyoruz ve bu sayede gemi ve tır bilgilerine ulaşabiliyoruz.

Ödev No: 2	Tarih 12.12.2023	
------------	------------------	--

KAYNAKÇA

Metin Tabanlı Yapay Zeka Modeli ChatGPT(GPT-3.5
mimarisi) W3Schools. <https://www.w3schools.com/>
Stack Overflow. <https://stackoverflow.com/>

Ödev No: 2	Tarih 12.12.2023	
------------	------------------	--