



T.C

**KOCaeli SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ LİSANSÜSTÜ
EęİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİęİ PROGRAMI**

ÖDEV KONUSU

YZM217-BLM217 Programlama Lab 1 - Ödev 2

Hazırlayanlar

ÖęRENCİ ADI: AHMET CAN BOSTANCI-220501031

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

ÖęRENCİ ADI: DİLA SERAY TEGÜN-220501022

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

DERS SORUMLUSU

Prof. Dr. H. Tarık DURU

TARİH 14.12.2023

İÇİNDEKİLER

ÖZET.....	2
GİRİŞ.....	2
KAYNAKÇA.....	13

ÖZET

Ödev yapılırken kullanıcının bizden istediği talepler doğrultusunda yazmaya özen gösterd.Bu süreçte Çeşitli kütüphaneler ,algoritmalar,fonksiyonlar ,nesneler, classlar yazılmış olup kod yazılırken faydalanılmıştır.

GİRİŞ

```
import time
import ast
from termcolor import *
```

Liman otomasyonunu gerçekleştirirken 3 tane kütüphane kullandık;

Time modülü çıktıları daha net görebilmesi için programı yavaşlatmak için kullanılmıştır.

Şuan ki ayarda program 359 saniyede sonlanmaktadır.(Time modülü kullanılmadığı zaman 13 saniyede program sona erer ve çıktıları takip edemezsiniz.)

Ast modülü **belirlisatırokuyucu** fonksiyonunda okunan stringleri liste tipine dönüşümünü sağlar. Örn: "[1,2,3,4] ".

Termcolor kütüphanesi olaya göre renkli çıktıları sağlamanıza olanak tanıyor.

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

```
def belirlisatırokuyucu(file,satır):  
    dosya=open(file,"r",encoding="utf-8")  
  
    for i in range(satır-1):  
        dosya.readline()  
  
    satırverisi=dosya.readline().strip("\n")  
    satırverisi=ast.literal_eval(satırverisi)  
    dosya.close()  
    return satırverisi
```

Kodumuzdaki **belirlisatırokuyucu** fonksiyonunun işlevi bir txt dosyasındaki sadece belirtilen satırı okuyup liste tipinde döndürmesidir.

```
def belirlisatıryazıcı(dosya_adı, satır, veri):  
  
    with open(dosya_adı, 'r', encoding='utf-8') as dosya:  
        satirlar = dosya.readlines()  
  
    if len(satirlar) >= satır:  
        del satirlar[satır - 1]  
  
    satirlar.insert(satır - 1, str(veri) + '\n')  
  
    with open(dosya_adı, 'w', encoding='utf-8') as dosya:  
        dosya.writelines(satirlar)
```

Belirlisatıryazıcı fonksiyonunun işlevi sadece belirtilen satır numarasına listeyi yazdırmaktır.

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

```

class istif:
    def __init__(self,ulke):

        self.ulke=ulke
        self.istif1=0
        self.istif2=0

    def bos_mu(self):
        return self.istif1==0 and self.istif2==0

    def yuk_ekle(self, eklencek_yuk):
        yukkontrol=self.istif1+self.istif2
        if yukkontrol+eklencek_yuk<=1500:

            for i in range(int(eklencek_yuk)):
                if self.istif1<750:
                    self.istif1+=1
                elif self.istif2<750:
                    self.istif2+=1
                if self.istif1==750:
                    print(colored("1.ci istif alanı doldu yükler 2.ci istif alanına yönlendiriliyor.", "red"))

            else:
                print("istif alanları kapasite yetersiz")

```

İstif classının yazılma amacı:

İstif alanındaki yükleri yönetmek yük giriş çıkış işlemleri için yazılmıştır.

bos_mu fonksiyonu istif alanlarının boş mu dolu mu olduğunu kontrol eder.

yuk_ekle fonksiyonu istif alanında eklenecek yük istif alanına sığar mı diye kontrol eder sığarsa yükü ekler sığmaz ise kapasite yetersiz uyarısı yazdırır.

```

def yuk_cikar(self,cıkanyuk):
    cıkacakyuk=cıkanyuk
    if not self.bos_mu():

        for i in range(int(cıkanyuk)):
            if self.istif2!=0:
                self.istif2-=1
            else:
                if self.istif1!=0:
                    self.istif1-=1
                else:
                    print("çıkacak bu kadar yük yok")

        else:
            print("istif alanı boş.")
    def toplam_yuk(self):
        return self.istif1+self.istif2
    def yeteri_yuk_var mı(self,cekilecekyuk):
        return cekilecekyuk<=self.istif1+self.istif2

```

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

yuk_cikar fonksiyonu istif alanından yük çıkarır stack yapısı gibi çalışır yükler ilk istif2 alanından(yük varsa) sonra istif1 alanından çıkarılır.

toplam_yuk fonksiyonu istif alanlarının anlık yük sayısını döndürür.

yeteri_yuk_varmi fonksiyonu çıkarılacak yükün istif alanında var mı yok mu olduğunun döndürür.

```
class TIR():
    def __init__(self,zaman,plaka):

        self.veri=tirdict[(zaman,plaka)]
        self.zaman=zaman
        self.plaka=plaka
        self.ulke=self.veri[0]
        self.yuk_kapasite=self.veri[1]
        self.yuk_miktari=self.veri[2]
        self.maliyet=self.veri[3]

    def yukbosmu(self):
        if self.yuk_miktari==0:
            return True
        else:
            return False
```

TIR classının yazılma amacı:

Her farklı tırı verilen parametrelere göre ayırt eder. bir tırı diğerlerinden ayırt eden faktörler

Zaman ve tır plakasıdır. Zaman ve tır plakası aynı bir tır yoktur. TIR classı verilen parametreleri

Göre tırı seçer ve diğer bilgileri erişmeye sağlar.(ülke,yük kapasitesi,yük miktarı,maliyet). Buda

kodun

okunabilirlik ve kolaylığı açısından avantaj sağlar.

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

```

class GEMİ():

    def __init__(self,numara,ilkcalıstırma):
        time.sleep(0.05)
        self.veri=gemidict[numara]
        self.kapasite=self.veri[1]
        if ilkcalıstırma:

            self.numara=numara
            self.zaman=self.veri[0]

            self.ulke=self.veri[2]

            self.yük=0
            self.min=self.kapasite*(95/100)
            veri=[numara,self.zaman,self.yük,self.min,self.ulke]
            belirlisatıryazıcı("gemidurumu.txt",int(numara),veri)
        else:
            veri=belirlisatırokuyucu("gemidurumu.txt",int(numara))

            self.numara=veri[0]
            self.zaman=veri[1]
            self.yük=veri[2]
            self.min=veri[3]
            self.ulke=veri[4]

    def gitmeyehazırmı(self):
        return self.yük>self.min
    def yük_ekle(self,yuk):
        self.yük+=yuk
        veri=[self.numara,self.zaman,self.yük,self.min,self.ulke]
        belirlisatıryazıcı("gemidurumu.txt",int(self.numara),veri)

```

GEMİ classının yazılma amacı:

Gemileri numarasına göre ayırt etmeye sağlar aynı numaralı başka gemiler yoktur.

Parametre olarak gemi numarası verilir ve zaman(self.zaman),anlık yük(self.yük), geminin harekete geçmesi gereken minimum yük (self.min), geminin gideceği ülke (self.ulke) elde edilir.

Gemi ilkcalıştırma parametresi alır True veya False değeri verilebilir.

Gemi ile ilk nesne oluştururken True ile oluşturuyoruz. Buda gemidurumu.txt dosyasına gemi bilgilerini yazıyor. Sırayla [numara,zaman,anlık yük,harekete geçmesi için gereken minimum yük, gideceği ülke]. Aynı gemi birkez daha seçildiğinde False değeri ile çalıştırıyoruz.

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

Gemi bilgilerini bir dosyaya kaydetmemizin temel amacı gemi bilgilerini hatırlamaktır.Özellikle geminin anlık yük durumunu hatırlamak için yazılmıştır.

Gitmeyehazırımı fonksiyonu geminin gitmeye hazır olup olmadığını kontrol eder ona göre True ,False Değer döndürür.

Yük_ekle fonksiyonu gemiye yük ekler ve gemidurumu.txt listesindeki anlık yük verisini günceller.

```
def plakasıralama(a):  
    return int(a[1][9:])
```

plakasıralama fonksiyonu amacı:

olaylarverisi listesi bir iç içe liste yapısıdır. [[6,41_kostu_616,Neverland,1,0,20,20000],
[6,41_kostu_616,Neverland,1,0,20,20000]] formatında tutulur.
Plaka sayısına göre küçükten büyüğe doğru sıralamak için kullanır.

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

```

140 tırsorgudict=dict()
141
142 olaylarverisi=[]
143 olaylar=open("olaylar.csv","r",encoding="ISO-8859-9")
144 tırdict={}
145 for satır in olaylar:
146     satırverisi=[]
147     anlıkveri=satır.split(",")
148     olaylarverisi.append([i.strip("\n") for i in anlıkveri])
149 olaylar.close()
150 olaylarverisi.pop(0) # başlıkların silinmesi
151 for sorgutır in olaylarverisi:
152     tırsorgudict[(str(sorgutır[0]),str(sorgutır[1]))]=sorgutır[2:]
153
154
155 for index in olaylarverisi:
156     if index[3]=="1":
157         tırdict[(int(index[0]),index[1])]=[index[2],20000,int(index[5]),index[6]]
158     else:
159         tırdict[(int(index[0]),index[1])]=[index[2],30000,int(index[5]),index[6]] # zaman ve plakaya karşılık : ülke ,yük kapasite,üzerindeki,maliyet
160

```

Yukarıdaki kod parçacığının açıklaması:

Tırsorgudict adında bir sözlük yapısı oluşturuyoruz. Daha sonra tırları sorgulamak için kullanacağız.

Tırdict sözlüğü oluşturuldu.(daha sonradan classlara veri sağlaması için)

Olaylar.csv dosyasını okumak için open fonksiyonu kullanılmıştır. Düzgün okuma yapılabilmesi için "ISO-8859-9" formatı ile açtık.

Dosya okumak için for döngüsü açtık ve sonunda elde edilen yapı olan olaylarverisi listesi

[[6,41_kostu_616,Neverland,1,0,20,20000], [6,41_kostu_616,Neverland,1,0,20,20000]] yapısına sahip bir liste oldu.

Olaylar.csv dosyasının ilk satırındaki sütun başlıklarını silmek için olaylarverisi.pop(0) kullandık.

(İşlem hatalarına sebep olmamak amacıyla).

olaylarverisi listesi for döngüsüne alındı tırsorgudict sözlüğüne ekleme işlemleri gerçekleştirildi.

Zaman ve plaka (zaman,plaka) key değerine karşılık [ulke,20 ton adet,30 ton adet,yük miktarı,maliyet] value değeri.

Olaylarverisi tekrardan tırdict sözlüğüne veri eklemek için for döngüsüne alındı.

(zaman,plaka) key değerine karşılık [ülke,yük kapasitesi,üzerindeki yük,maliyet] formatında

20 ton adet miktarı 1 olanlarda direk yük kapasitesini 20000 olmayanlarda 30000 değeri şeklinde değer ekledik.

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--


```
163 gemilerverisi=[]
164 gemisorgudict=dict()
165 gemiler=open("gemiler.csv","r",encoding="ISO-8859-9")
166 for satır in gemiler:
167     satırverisi=[]
168     anlıkveri=satır.split(",")
169     gemilerverisi.append([i.strip("\n") for i in anlıkveri])
170 gemiler.close()
171 gemilerverisi.pop(0) # başlıkların silinmesi
172
173 for gemisorgu in gemilerverisi:
174     gemisorgudict[gemisorgu[1]]=gemisorgu[0],gemisorgu[2],gemisorgu[3]]
175 gemidict=dict()
176 for index in gemilerverisi:
177     gemidict[index[1]]=int(index[0]),int(index[2]),index[3] # gemi numarasına karşılık : zaman kapasite ülke
```

Gemilerverisi listesi gemiler.csv dosyasındaki verileri tutmak için atandı.

Gemisorgudict sözlüğü gemi bilgilerini sorgulamak için yazılmıştır.

Gemidict sözlüğü oluşturuldu. (daha sonra classlara veri sağlayacak)

Satır 166'dan satır 171'e kadar olaylarverisi ile aynı işlemler yapılmıştır

Satır 173'deki for döngüsünde tırdict sözlüğüne benzer veri eklemesi yapmaktadır. (gemi numarası key değerine karşılık [zaman,kapsite,ülke] value değeri).

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

```
180 tır_dict_keys=tirdict.keys()
181 gemi_dict_keys=gemidict.keys()
182 zaman_listesi=[]
183 for zaman in tır_dict_keys:
184     if not (int(zaman[0]) in zaman_listesi):
185         zaman_listesi.append(int(zaman[0]))
186
187 geminumara_iter=iter(gemidict.keys())
188
189
190 neverland_istifalanı=istif("Neverland")
191 mordor_istifalanı=istif("Mordor")
192 lilli_istifalanı=istif("Lilliputa")
193 ocean_istifalanı=istif("Oceania")
194 anlık_gemi=GEMİ(next(geminumara_iter),True)
195 bekleyengemiler=[]
196 yukindirentırlar=[]
197
```

Satır 180 ve 181 de. Oluşturduğumuz sözlüklerin key değerlerini listeye attı.

Satır 183 ve 185 deki for döngündeki amaç birbirinden farklı zamanları zaman_listesine atar.

Satır 187’de iterable bir nesne oluşturuyoruz. Amaç her next() fonksiyonu çağrıldığında gemidict.keys()

İçindeki verileri teker teker döndürmesi için yazılmıştır.

Satır 190’dan satır 193’e kadar ülkelerin yüklerinin yönetmek için istif objeleri oluşturulmuştur.

194’de anlık gemi nesnesi ilerdeki for döngülerinde her defasında başka bir gemi seçilmesini sağlar.

Bekleyengemiler listesi yükleri dolmamış gemiler için atanmıştır.

Yukindirentırlar listesi yükünü indirmiş tırlar için atanmıştır.

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

```

198 for eszaman in zaman_listesi:
199
200     eszamanlirtirlist=[]
201     for eszamanlr_tir in tir_dict_keys: # aynı zamanları bir döngüye atmak için
202         eszamanlirtir=TIR(eszamanlr_tir[0],eszamanlr_tir[1])
203         if eszaman==eszamanlirtir.zaman:
204             eszamanlirtirlist.append(eszamanlr_tir)
205
206
207
208     eszamanlirtirlist.sort(key=plakasiralama)
209     for tirlar in eszamanlirtirlist:
210         aynizamanlirtir=TIR(tirlar[0],tirlar[1])
211         if ocean_istifalani.toplam_yuk()+lilli_istifalani.toplam_yuk()+mordor_istifalani.toplam_yuk()+neverland_istifalani.toplam_yuk()<=1500:
212             if not (tirlar[0],tirlar[1]) in yukindirentirlar:
213                 print(colored("{} zamanında {} plakalı tir {} ülkesine gitmek üzere {} ton yük indirdi maliyet:{}".format(aynizamanlirtir.zaman,aynizamanlirtir.plaka,aynizamanlirtir.ulke,aynizamanlirtir.yukindirentirlar.append((tirlar[0],tirlar[1]))
214
215
216             if anlik_gemi.gitmeyehazirim():
217                 print(colored("{} gemisi limandan ayrıldı. {}'e gidiyor".format(anlik_gemi.numara,anlik_gemi.ulke,"red"))
218                 anlik_gemi=GEMİ(next(geminumara_iter),True)
219
220             if aynizamanlirtir.zaman>anlik_gemi.zaman and aynizamanlirtir.ulke==anlik_gemi.ulke:
221                 print(colored("{} zamanında {} numaralı gemiye {}ton yük yüklendi. gemi {} gitmek üzere bekliyor anlık yük: {} harekete geçeceği yük:{}".format(aynizamanlirtir.zaman,anlik_gemi.numara,aynizamanlirtir.yuk_ekle(aynizamanlirtir.yuk_miktari)
222                 anlik_gemi.yuk_ekle(aynizamanlirtir.yuk_miktari)
223             else:
224                 print(colored("{} yükü {} gemisi gelene kadar istif alanında bekletiliyor".format(aynizamanlirtir.plaka,aynizamanlirtir.ulke),"yellow"))
225                 if aynizamanlirtir.ulke=="Oceania":
226                     ocean_istifalani.yuk_ekle(aynizamanlirtir.yuk_miktari)
227                 elif aynizamanlirtir.ulke=="Lilliputa":
228                     lilli_istifalani.yuk_ekle(aynizamanlirtir.yuk_miktari)
229
230                 elif aynizamanlirtir.ulke=="Mordor":
231                     mordor_istifalani.yuk_ekle(aynizamanlirtir.yuk_miktari)
232                 elif aynizamanlirtir.ulke=="Neverland":
233                     neverland_istifalani.yuk_ekle(aynizamanlirtir.yuk_miktari)

```

```

235 ~     else:
236
237 ~         if ocean_istifalani.toplam_yuk()+lilli_istifalani.toplam_yuk()+mordor_istifalani.toplam_yuk()+neverland_istifalani.toplam_yuk()<=1500:
238
239             print(colored("istif alanı kapasitesi doldu"),"red")
240
241             bekleyengemiler.append(anlik_gemi.numara)
242             anlik_gemi=GEMİ(next(geminumara_iter),True)
243
244             bekleyengemilerfor=bekleyengemiler
245 ~             for bekleyengemi in bekleyengemilerfor:
246                 anlik_gemi=GEMİ(bekleyengemi,False)
247 ~                 if anlik_gemi.gitmeyehazirim():
248                     print(colored("{} gemisi {} gitmek üzere limandan ayrıldı.".format(anlik_gemi.numara,anlik_gemi.ulke),"magenta"))
249                     bekleyengemiler.remove(bekleyengemi)
250 ~                 if anlik_gemi.ulke=="Oceania":
251 ~                     if ocean_istifalani.yeteri_yuk_varmi(anlik_gemi.min-anlik_gemi.yuk):
252                         print(colored("{} gemisine istif alanından {} ton yük yüklendi ve {}e gitmek üzere limandan ayrıldı. ".format(anlik_gemi.numara,anlik_gemi.min-anlik_gemi.yuk,anlik_gemi.ulke),
253                         ocean_istifalani.yuk_cikar(anlik_gemi.min-anlik_gemi.yuk)
254                         anlik_gemi.yuk_ekle(anlik_gemi.min-anlik_gemi.yuk)
255
256                         bekleyengemiler.remove(bekleyengemi)
257
258
259 ~                 elif anlik_gemi.ulke=="Lilliputa":
260 ~                     if lilli_istifalani.yeteri_yuk_varmi(anlik_gemi.min-anlik_gemi.yuk):
261                         print(colored("{} gemisine istif alanından {} ton yük yüklendi ve {}e gitmek üzere limandan ayrıldı. ".format(anlik_gemi.numara,anlik_gemi.min-anlik_gemi.yuk,anlik_gemi.ulke),
262                         lilli_istifalani.yuk_cikar(anlik_gemi.min-anlik_gemi.yuk)
263                         anlik_gemi.yuk_ekle(anlik_gemi.min-anlik_gemi.yuk)
264
265                         bekleyengemiler.remove(bekleyengemi)

```

```

269 ~                 elif anlik_gemi.ulke=="Mordor":
270 ~                     if mordor_istifalani.yeteri_yuk_varmi(anlik_gemi.min-anlik_gemi.yuk):
271                         print(colored("{} gemisine istif alanından {} ton yük yüklendi ve {}e gitmek üzere limandan ayrıldı. ".format(anlik_gemi.numara,anlik_gemi.min-anlik_gemi.yuk,anlik_gemi.ulke),
272                         mordor_istifalani.yuk_cikar(anlik_gemi.min-anlik_gemi.yuk)
273                         anlik_gemi.yuk_ekle(anlik_gemi.min-anlik_gemi.yuk)
274
275                         bekleyengemiler.remove(bekleyengemi)
276
277
278 ~                 elif anlik_gemi.ulke=="Neverland":
279 ~                     if neverland_istifalani.yeteri_yuk_varmi(anlik_gemi.min-anlik_gemi.yuk):
280                         print(colored("{} gemisine istif alanından {} ton yük yüklendi ve {}e gitmek üzere limandan ayrıldı. ".format(anlik_gemi.numara,anlik_gemi.min-anlik_gemi.yuk,anlik_gemi.ulke),
281                         neverland_istifalani.yuk_cikar(anlik_gemi.min-anlik_gemi.yuk)
282                         anlik_gemi.yuk_ekle(anlik_gemi.min-anlik_gemi.yuk)
283
284                         bekleyengemiler.remove(bekleyengemi)

```

Yukarıdaki son 3 kodu özetleyerek açıklamak istersek:

Tırlar her döngüde eşzamanlı gelen tırları eşzamanlı tırlar listesine atar ve satır 208’de olduğu gibi plakalarını küçükten büyüğe doğru sıralar. Bir for döngüsü daha açılır eşzamanlı tırlar içindeki veriler ile her defasında TIR classında seçim yapılır. Ve yükler küçük plakalı tırdan büyük plakalı tıra doğru sırayla indirir. Her defasında istif alanın dolup dolmadığı kontrolü yapılır. Yük indirirken yükindirentırlar listesine ekleme yapar (Her defasında yükindirentırlar listesine aynı eleman eklenemez şeklinde şart kontrolü yapılır). Her for döngüsünde gemi gitmeye hazır mı kontrol edilir. Gemi hazırsa limandan ayrıldı çıktısı gönderir ve bir sonraki gemiye geçer. Her for döngüsünde tırın zamanı geminin zamanından fazla veya eşitse ve aynı zamanda tırın yükünün ülkesini ve gemisinin ülkesi eşit ise gemiye yük ekler. Eğer eşit değilse yük ilgili istif alanına bırakılır. Ve seçilen gemiyi bekleyen gemiler listesine ekler. İstif alanın yükü dolarsa istif alanın yükünü boşaltmak için bekleyen gemiler listesindeki gemilere istif alanından uygun yükler ile yükleme yapılır. Ve hazır olan gemiler bekleyen gemiler listesinden çıkarılır ve limandan ayrılır çıktısı döner.

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--

KAYNAKÇA

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

Ödev No: 2	Tarih 14.12.2023	
------------	------------------	--