



**T.C**

**KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİęİ**

**LİMAN OTOMASYONU**

**Hazırlayan Öğrenciler:**

**Enes KÜÇÜK-220501017**

**<https://github.com/EnessKucukk>**

**İbrahim Bener KARACA-220501019**

**<https://github.com/BenerKaraca>**

**DERS SORUMLUSU**

**ÖĞR. GÖR. PROF. DR. Hüseyin Tarık DURU**

---

## TARİH

## İÇİNDEKİLER

1. ÖZET .....	3
2. GİRİŞ .....	3
3. YÖNTEM .....	5
3.1 Class kullanımı .....	5
3.2 Fonksiyonlar .....	7
3.3 Kütüphane .....	7
5. SONUÇ VE ÖĞRENİLEN DERSLER .....	10
6. KAYNAKÇA .....	10

Ödev No: 1	Tarih 11.12.2023	2/10
------------	------------------	------

## 1. ÖZET

Bir Liman otomasyonu yapıldı. Bu otomasyon gemileri ve olayları toplu bir şekilde simüle eder ve bunları kullanıcaya bildirir.

## 2. GİRİŞ

Bu Ödevde bizden bir liman otomasyonu istendi. Ödevdeki gerekli şartlar:

□ Her t anında ilk olarak TIRlar yüklerini indirecektir ve ardından yükler gemilere yüklenecektir.

□ Aynı anda “olaylar.csv” dosyasında yer alan tüm TIRlar indirme yapabilir fakat plaka numaralarına göre sıralı olarak indirme işlemi yapılmalıdır.

□ TIRlar yüklerini indirirken plaka numarası küçük olan TIR yükünü ilk olarak indirecektir. Plaka numarası küçükten büyüğe doğru olacak şekilde diğer TIRlar yüklerini indirmeye devam edecektir.

□ İlk TIRın plakası örnek olarak “41\_kostu\_001” olabilir ve TIRlar plakaları kullanılarak adlandırılabilir.

TIRların plakalarını küçükten büyüğe doğru sıralamada kullanabilmek amacıyla “001”li kısım kullanılabilir. Aynı kısım, “002, 003, 004, ...” şeklinde arttırılarak diğer TIRlar için yeni plakalar oluşturulabilir.

□ Limandan sadece 4 farklı ülkeye yük taşınmaktadır. Bunlar Mordor, Neverland, Lilliputa ve Oceania ülkeleridir.

□ Gemiler yüklerini yüklerken numarası küçük olan gemi yükünü ilk önce yükleyecektir. Gemilerin

numaralandırılması örnek olarak “001, 002, 003, ...” şeklinde olabilir.

□ Aynı anda sadece tek bir gemi yüklenebilir. Yüklenen gemi en küçük numaralı gemidir. Gemi kapasitesinin en az %95’i dolduğu anda limanı terk eder ve yükleme sırası limandaki en küçük numaralı gemiye geçer. Limana gelen gemilerin numaraları sıralı şekilde artar.

Ödev No: 1	Tarih 11.12.2023	3/10
------------	------------------	------

---

☐ TIR'lar sadece iki farklı tonajdaki konteyner içinde kuru yük taşıyabilmektedir. Bu konteynerler her zaman tam yüklü olarak gelmektedir ve kapasiteleri 20 veya 30 tondur.

☐ Limanda 2 istif alanı ve tek bir vinç mevcuttur. Bu vinç ile önce t anındaki tüm yükler indirilmektedir.

İndirilen yükler her zaman 1 numaralı istif bölgesine ve üst üste olacak şekilde yerleştirilir.

☐ Bir istif alanının kapasitesi 750 tondur. İstif alanları dolduğunda istif alanının dolu olduğuna dair geribildirim gelmelidir. İstif alanı boşaldığında, boş olduğuna dair geri bildirim gelmelidir.

☐ Alanda yük yoksa ve t anında gelen gemi varsa gemi beklemeye devam edecektir. İstif alanı doluyorsa

limana gelmiş olan TIR'lar beklemeye devam edecektir ve aynı zamanda, zaman ilerlemeye devam edecektir.

☐ Yükleme işlemi için t zamanında en fazla 20 vinç işlemine izin verilir. Her bir vinç işlemi yükü ya bir istif alanından diğerine taşır veya yükü gemiye yükler.

☐ Gemilerin kapasiteleri 250, 300, 300 veya 500 tondur.

☐ TIR kapasiteleri iki çeşittir. Bu kapasiteler sabittir ve 20 veya 30 tondur.

Kodda istenen şartlar ise şunlar:

“olaylar.csv” adında bir dosya kullanılması istenmektedir. Bu dosyada gelen yüklerin listesi yer almaktadır. Dosyadaki bilgiler okunarak ilgili yük indirme-yükleme işlemlerinin tamamlanması istenmektedir.

☐ “gemiler.csv” adında bir dosya kullanılması istenmektedir. Bu dosyada limana gelecek gemilerin listesi yer almaktadır. Dosyadaki bilgiler okunarak ilgili yük indirme-yükleme işlemlerinin tamamlanması istenmektedir.

☐ TIR ve gemi adında sınıflar oluşturulmalıdır. Oluşturulan gemiler gemi sınıfına ve TIRlar TIR sınıfına ait olmalıdır.

☐ Oluşturulan her gemi ve TIR için sözlük veri tipinde birer değişken oluşturulmalıdır. Sözlük veri tipindeki bu değişkende yükün götürüleceği ülke, 20 tonluk konteynır adedi, 30 tonluk konteynır adedi ve yükün miktarı (ton cinsinden), yükün maliyeti (Türk Lirası cinsinden) bilgileri yer almalıdır. Bu değişken kullanılarak istendiğinde tır veya geminin yük bilgisine erişilebilmelidir.

Ödev No: 1	Tarih 11.12.2023	4/10
------------	------------------	------

### 3. YÖNTEM

Bu projede kullandığımız method başlıkları kısaca şunlardır:

#### 3.1 Class kullanımı

```
4. class Gemi:
    durum: dict = {}

    def __init__(self, gemi_adi, kapasite, gidecek_ulke):
        self.info = {
            "gemi_adi": gemi_adi,
            "kapasite": kapasite,
            "gidecek_ülke": gidecek_ulke
        }
        self.sira = gemi_adi
        self.mevcut_kapasite = 0

    def yukleme(self, yuk):
        self.mevcut_kapasite += yuk

    def gitmeye_hazirlar(self):
        return self.mevcut_kapasite / self.info["kapasite"] >= 0.95

    @staticmethod
    def okuma(path):
        a = pd.read_csv(path, encoding="cp1254")
        maksimum_t = 0
        for i in range(len(a)):
            gemi = a.iloc[i]
            Gemidurum = Gemi(
                a.iloc[i]["gemi_adi"],
                a.iloc[i]["kapasite"],
                a.iloc[i]["gidecek_ülke"]
            )
            if Gemi.durum.get(gemi["geliş_zamanı"]) is None:
                Gemi.durum[gemi["geliş_zamanı"]] = [Gemidurum]
            else:
                Gemi.durum[gemi["geliş_zamanı"]].append(Gemidurum)
            maksimum_t = max(maksimum_t, gemi["geliş_zamanı"])
        return maksimum_t
```

Sınıf Oluşturma: Gemi sınıfı, gemi özelliklerini ve durumunu içeren bir sınıftır.

Durum Sözlüğü: durum adlı bir sınıf değişkeni, gemi durumlarını geliş zamanına göre saklamak için bir sözlüktür.

İnit Fonksiyonu: Gemi sınıfının başlatıcı fonksiyonu, gemi özelliklerini ve durumunu başlatır.

Yükleme Fonksiyonu: yukleme fonksiyonu, gemiye yük ekler ve mevcut kapasiteyi günceller.

Hazır Durum Kontrolü: gitmeye\_hazirlar fonksiyonu, geminin gitmeye hazır olup olmadığını kontrol eder.

CSV Okuma Fonksiyonu (okuma): Verilen CSV dosyasını okur ve gemi bilgilerini içeren bir DataFrame oluşturur.

Ödev No: 1	Tarih 11.12.2023	5/10
------------	------------------	------

Gemi Oluřturma: Her bir satırı döngü içinde iřler ve gemi örnekleri oluřturarak durum sözlüğüne ekler.

Maksimum Geliř Zamanı: En sonunda maksimum geliř zamanını döndürür.

```
class Tir:
    durum: dict = {}

    def __init__(self, plaka: str, ulke: str, ton20: int, ton30: int,
yuk_miktari: int, maliyet: int):
        self.info = {
            "plaka": plaka,
            "ülke": ulke,
            "20_ton_adet": ton20,
            "30_ton_adet": ton30,
            "yük_miktari": yuk_miktari,
            "maliyet": maliyet
        }
        self.sira = plaka.split("_")[-1]

    @staticmethod
    def okuma(path):
        a = pd.read_csv(path, encoding="cp1254")
        maksimum_t = 0
        for i in range(len(a)):
            tir = a.iloc[i]
            Tirdurum = Tir(
                a.iloc[i]["tır_plakası"],
                a.iloc[i]["ülke"],
                a.iloc[i]["20_ton_adet"],
                a.iloc[i]["30_ton_adet"],
                a.iloc[i]["yük_miktari"],
                a.iloc[i]["maliyet"]
            )
            if Tir.durum.get(tir["geliř_zamanı"]) is None:
                Tir.durum[tir["geliř_zamanı"]] = [Tirdurum]
            else:
                Tir.durum[tir["geliř_zamanı"]].append(Tirdurum)
            maksimum_t = max(maksimum_t, tir["geliř_zamanı"])
        return maksimum_t
```

Sınıf Oluřturma: Tir sınıfı, TIR'ın özelliklerini ve durumunu içeren bir sınıftır.

Durum Sözlüğü: durum adlı bir sınıf değışkeni, TIR durumlarını geliř zamanına göre saklamak için bir sözlüktür.

İnit Fonksiyonu: \_\_init\_\_ fonksiyonu, TIR özelliklerini başlatır.

CSV Okuma Fonksiyonu (okuma): Verilen CSV dosyasını okur ve TIR bilgilerini içeren bir DataFrame oluřturur.

TIR Oluřturma: Her bir satırı döngü içinde iřler ve TIR örnekleri oluřturarak durum sözlüğüne ekler.

Maksimum Geliř Zamanı: En sonunda maksimum geliř zamanını döndürür.

Ödev No: 1	Tarih 11.12.2023	6/10
------------	------------------	------

## 3.2 Fonksiyonlar

```
4 if __name__ == "__main__":
    tir_max_t = Tir.okuma("olaylar.csv")
    gemi_max_t = Gemi.okuma("gemiler.csv")

    gemi_bekletme_bolgesi = []
    tir_bekletme_bolgesi = []
    istif_alani = [[], []]

    for t in range(1, max(tir_max_t, gemi_max_t) + 1):
        print(f"{t} tarihi için işlemler başlatıldı")
        vinc_kullanma_miktari = 0
        if Tir.durum.get(t) is not None:
            for tir in Tir.durum[t]:
                tir_bekletme_bolgesi.append(tir)
                print(f"\t\t\t tarihinde {tir.info['plaka']} plakalı tir
bekletme alanına geldi")
                tir_bekletme_bolgesi.sort(key=lambda tir: tir.sira)

            if Gemi.durum.get(t) is not None:
                for gemi in Gemi.durum[t]:
                    gemi_bekletme_bolgesi.append(gemi)
                    print(f"\t\t\t tarihinde {gemi.info['gemi_adi']} adlı
gemi bekletme alanına getirildi")
                    gemi_bekletme_bolgesi.sort(key=lambda gemi: gemi.sira)

                while istif_alan_yukleri(istif_alani[0]) < 750 and
vinc_kullanma_miktari < 20:
                    if len(tir_bekletme_bolgesi) == 0:
                        break
                    tir = tir_bekletme_bolgesi.pop(0)
                    yuk = tir.info["20_ton_adet"] * 20 + tir.info["30_ton_adet"]
* 30

                    if istif_alan_yukleri(istif_alani[0]) + yuk > 750:
                        tir_bekletme_bolgesi = [tir] + tir_bekletme_bolgesi
                        break
                    else:
                        istif_alani[0].append([yuk, tir.info["ülke"]])
                        vinc_kullanma_miktari += 1
                        print(
                            f"\t\t\t tarihinde {tir.info['plaka']} plakalı tir 1
nolu istif alanından {tir.info['ülke']}adlı ülkeye gidecek kargo
yukledi")

                        if istif_alan_yukleri(istif_alani[0]) == 750 and
vinc_kullanma_miktari < 20:
                            print(f"\t\t\t tarihinde 750 tonluk 1 nolu istif alanı
doldu")

                        while istif_alan_yukleri(istif_alani[1]) < 750:
                            if len(tir_bekletme_bolgesi) == 0:
                                break
                            tir = tir_bekletme_bolgesi.pop(0)
                            yuk = tir.info["20_ton_adet"] * 20 + tir.info["30_ton_adet"]
```

Ödev No: 1	Tarih 11.12.2023	7/10
------------	------------------	------

```

* 30
        if istif_alan_yukleri(istif_alani[1]) + yuk > 750:
            tir_bekletme_bolgesi = [tir] + tir_bekletme_bolgesi
            break
        else:
            istif_alani[1].append(yuk)
            vinc_kullanma_miktari += 1
            print(
                f"\t\t tarihinde {tir.info['plaka']} plakalı tır 2
nolu istif alanından {tir.info['ülke']} adlı ülkeye gidecek kargo
yükledi")

            if istif_alan_yukleri(istif_alani[1]) == 750:
                print(f"\t\t tarihinde 750 tonluk 2 nolu istif alanı
dolmuştur")

            index = 0
            while istif_alan_yukleri(istif_alani[0]) > 0 and index <
len(gemi_bekletme_bolgesi) and vinc_kullanma_miktari < 20:
                if len(gemi_bekletme_bolgesi) == 0:
                    break
                gemi = gemi_bekletme_bolgesi[index]
                gidilecek_ülke = gemi.info["gidecek_ülke"]
                kargo_listesi = list(filter(lambda x: x[1] ==
gidilecek_ülke, istif_alani[0]))
                for k in kargo_listesi:
                    gemi.yukleme(k[0])
                    vinc_kullanma_miktari += 1
                    istif_alani[0].remove(k)
                    print(
                        f"\t\t tarihinde {k[0]} tonluk kargo 1 nolu istif
alanında {gemi.info['gemi_adi']} gemisine yüklendi")
                    if gemi.gitmeye_hazirlar():
                        print(f"\t\t tarihinde {gemi.info['gemi_adi']} adlı
gemi {gidilecek_ülke} ülkesine doğru harekete geçti")
                        gemi_bekletme_bolgesi.pop(index)
                        index -= 1
                    break
                index += 1

            if istif_alan_yukleri(istif_alani[0]) == 0:
                print(f"\t\t tarihinde 1 nolu istif alanı boşaltıldı ")

            index = 0
            while istif_alan_yukleri(istif_alani[1]) > 0 and index <
len(gemi_bekletme_bolgesi) and vinc_kullanma_miktari < 20:
                if len(gemi_bekletme_bolgesi) == 0:
                    break
                gemi = gemi_bekletme_bolgesi[index]
                gidilecek_ülke = gemi.info["gidecek_ülke"]
                kargo_listesi = list(filter(lambda x: x[1] ==
gidilecek_ülke, istif_alani[1]))
                for k in kargo_listesi:
                    gemi.yukleme(k[0])
                    istif_alani[1].remove(k)

```



```
vinc_kullanma_miktari += 1
print(
    f"\t\t tarihinde {gemi.info['gemi_adi']} adlı gemi
2 nolu istif alanına {k[0]} tonluk kargo yükledi")
    if gemi.gitmeye_hazirlar():
        print(f"\t\t tarihinde {gemi.info['gemi_adi']} adlı
gemi {gidilecek_ulke} ülkesine doğru harekete geçti ")
        gemi_bekletme_bolgesi.pop(index)
        index -= 1
        break
    index += 1

if istif_alan_yukleri(istif_alani[1]) == 0:
    print(f"\t\t tarihinde 2 nolu istif alanı boşaltıldı")

print(f"{t} tarihli işlemler sonlandırıldı")
```

CSV Dosyalarından Veri Okuma:

Tır sınıfı için olaylar.csv dosyasından tır verileri okunur. Gemi sınıfı için gemiler.csv dosyasından gemi verileri okunur.

Bekleme Alanları ve İstif Alanı Oluşturma:

tir\_bekletme\_bolgesi: Tır bekletme bölgesini temsil eden liste.

gemi\_bekletme\_bolgesi: Gemi bekletme bölgesini temsil eden liste.

istif\_alani: İstifleme alanını temsil eden liste, iki alt alan içerir.

Zaman İçinde İşlemleri Gerçekleştirme:

Belirli zaman aralıklarında tır ve gemi işlemleri yapılır.

Tır ve gemiler bekletme bölgelerine eklenir ve belirli koşullara göre işlemler gerçekleştirilir.

İstif Alanına Yükleme İşlemleri:

istif\_alani listesine tır yükleri eklenir. İstif alanı kapasitesi ve kullanım sınırları kontrol edilir.

Gemiye Yükleme İşlemleri:

Gemi bekletme bölgesindeki gemilere, istif alanındaki kargolar yüklenir.

Belirli koşullar altında gemilerin harekete geçmesi sağlanır.

Çıktıları Ekrana Yazdırma:

İşlemlerin durumunu ve gerçekleşen olayları belirli bir formatta ekrana yazdırır.

## 4.2 Kütüphane

```
import pandas as pd
```

pandas kütüphanesi kullanılmıştır.

Ödev No: 1	Tarih 11.12.2023	9/10
------------	------------------	------

---

## 5 SONUÇ VE ÖĞRENİLEN DERSLER

Bu ödevimizde bir liman işletmeciliği yapar gibi düşünerek bunları koda döktük. Böylece bir otomasyon çalışması tasarladık. Bu ödev bize csv dosyalarının okuma ve analiz yeteneği kattı.

## 6 KAYNAKÇA

<https://muhammeddincer.com/pandas-dataframe-ve-kod-ornekleri/>

<https://www.codingtxt.com/blog/detail/pandas-kutuphanesi-hakkinda-detayli-inceleme-ve-kod-ornekleri-79748d49-9156-45#gsc.tab=0>

<https://teknoloji.org/pandas-nedir-nasil-kullanilir-python-kutuphanesi/>

[https://www.udemy.com/share/101Wcu3@uTC-B8ifun\\_TM1DoKWUiq1ExPdPRnOKqb7EBhUbuWraPkCn\\_nq6QBX7LC3Wk3SyC/](https://www.udemy.com/share/101Wcu3@uTC-B8ifun_TM1DoKWUiq1ExPdPRnOKqb7EBhUbuWraPkCn_nq6QBX7LC3Wk3SyC/)

<https://www.programiz.com/python-programming/methods/built-in/staticmethod>

<https://www.youtube.com/watch?v=ZyhVh-qRZPA&list=PL-osiE80TeTsWmV9i9c58mdDCSskIFdDS>

Ödev No: 1	Tarih 11.12.2023	10/10
------------	------------------	-------