T.C

KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAZILIM MÜHENDİSLİĞİ PROGRAMI

ÖDEV KONUSU
LİMAN OTOMASYONU

HAZIRLAYAN

ÖĞRENCİ ADI: Aybüke TÜRİDİ

Elif Beyza BEYAZ

ÖĞRENCİ NUMARASI: 220502005

220502033

GITHUB LİNKLERİ

Aybüke TÜRİDİ https://github.com/220502005

Elif Beyza BEYAZ https://github.com/beyzabeyaz0

DERS SORUMLUSU

Prof. Dr. H. Tarık DURU

TARİH

13.12.2023

İÇİNDEKİLER

1. ÖZET

1.a) Amaç:

Bu projenin temel amacı, bir limandaki yük indirme-yükleme süreçlerini otomasyonlaştırmaktır. Proje, tırların yük indirmesi ve gemilere yük yükleme süreçlerini modellenmiş bir ortamda gerçekleştirir. Aşağıda projenin ana amaçları özetlenmiştir:

Yük İndirme ve Yükleme Simülasyonu: Limanda bulunan TIR'lar ve gemiler arasındaki yük indirme ve yükleme işlemlerini simüle eder.

Plaka ve Numaralandırma Sırası: Tırlar plakalarına göre sıralı olarak indirme yapar, gemiler ise numaralarına göre sıralı olarak yükleme yapar. Bu sayede, süreçlerin belirli bir sıra içinde gerçekleşmesi sağlanır.

Vinç İşlemleri ve İstif Alanları: Limandaki vinçler, yükleri istif alanlarına taşır ve belirli kapasitelerdeki istif alanlarını kullanır. Vinç işlemleri ve istif alanlarındaki kapasite durumu simüle edilir.

Dosya Tabanlı Veri Girişi: "olaylar.csv" ve "gemiler.csv" adlı dosyalar aracılığıyla gerçekleştirilen veri girişi sayesinde, sistem dinamik verilerle çalışılabilir.

Sınıflar ve Veri Yapıları: Tır ve gemi sınıfları ile sözlük veri yapıları kullanılarak, her bir aracın özellikleri ve yük bilgileri düzenli bir şekilde yönetir.

1.b) Yük İndirme-Yükleme Otomasyon Sistemi Özeti:

İşleyiş aşağıdaki başlıklar altında özetlenmiştir:

- **TIR ve Gemi İşleyişi:**

  Tırlar plakalarına göre sıralı olarak indirme yapar.

  Gemiler numaralarına göre sıralı olarak yükleme yapar.

  Gemilerin kapasitesi ve tırların tonajı gibi özellikler sabittir.

- **Senaryo İşleyişi:**

  Her an TIR'lar yüklerini plaka numaralarına göre sıralı olarak indirir.

  İndirilen yükler, gemilere numaralarına göre sıralı olarak yüklenir.

  Limandaki 4 farklı ülkeye yük taşınmaktadır.

  Bir gemi en az %95 kapasite dolduğunda limanı terk eder ve yükleme sırası limandaki en küçük numaralı gemiye geçer.

- **Dosya Kullanımı:**

"olaylar.csv" dosyası, TIR'ların yüklerini içermektedir.

"gemiler.csv" dosyası, limana gelecek gemilerin listesini içermektedir.

Bu dosyalardaki bilgiler okunarak yük indirme-yükleme işlemleri gerçekleştirilir.

- **Sınıflar ve Veri Yapıları:**

'TIR' ve 'GEMİ' sınıfları oluşturulmuştur.

Her bir tır ve gemi için sözlük veri tipinde birer değişken kullanılmıştır.

Bu değişkenler, yükün götürüleceği ülke, konteyner adetleri, yük miktarı ve maliyet bilgilerini içermektedir.

2. GİRİŞ

Liman Otomasyonu projesi, belirli bölümlerden oluşan bir Python uygulamasıdır. Bu bölümler: sınıflar, temel görev sınıfı, CSV dosya kullanımı ve veri okuma, fonksiyonlar ve ana program olarak ayrılmıştır. Projenin genel amacı, bir limandaki yük indirme-yükleme süreçlerini simüle etmek ve bu süreçleri etkin bir şekilde yönetmek üzere tasarlanmıştır.

1. Sınıflar
   Proje, tır ve gemi sınıflarını içermektedir. Bu sınıflar, ilgili araçların özelliklerini ve davranışlarını temsil etmektedir. Tır ve gemi nesneleri, bu sınıflar kullanılarak oluşturulmaktadır.

2. Temel görev sınıfı
   LimanOtomasyonu adlı temel görev sınıfı, tır ve gemi nesnelerini yöneterek limandaki yük transfer operasyonlarını simüle etmektedir. Bu sınıf, tır ve gemi verilerini CSV dosyalarından okuyarak, yük indirme ve yükleme işlemlerini düzenler.

3. Csv dosya kullanımı ve veri okuma
   Projede, "olaylar.csv" ve "gemiler.csv" adlı iki CSV dosyası kullanılmaktadır. Bu dosyalardan alınan veriler, program tarafından okunarak tır ve gemi nesnelerinin oluşturulmasını sağlar.

4. Fonksiyonlar
   Proje, istif alanlarına yük bırakma, tırlardan gemilere yükleme, sözlük oluşturma gibi fonksiyonları içermektedir. Bu fonksiyonlar, liman operasyonlarının düzenlenmesi ve yük transferinin etkin bir şekilde gerçekleştirilmesi için tasarlanmıştır.

5. Ana program
   Projenin ana programı, LimanOtomasyonu sınıfının örneklenmesi ve ilgili fonksiyonların çağrılması üzerine kurulmuştur. CSV dosyalarından veri okuma, yük transfer işlemleri ve kullanıcıya bilgi sunma gibi temel görevler, ana program içerisinde gerçekleştirilmektedir.

3. YÖNTEM

Proje geliştirme sürecinde belirli görev paylaşımları yapılıp yapılanlar hakkında grup içi devamlı bir bilgilendirme bulunmuştur.
Python üzerinden yapılan bu otomasyon uygulamasında belirli kod bölümleri bulunmaktadır. Bu bölümler ve bölümlerden kullanılan özellikler, yöntemler aşağıdakiler gibidir.

1. TIR, GEMİ, OTOMASYON SINIFLARI
   -Tır
   Projenin temel yapı taşları olan tır, otomasyon ve gemi sınıfları tasarlandı. Her bir sınıf, ilgili aracın özelliklerini içeren veri alanları ve aracın davranışlarını temsil eden metotları içermektedir. Bu tasarım sayesinde her aracın özel özelliklere sahip olması ve yönetilmesi mümkün hale gelmiştir.

```
class Tir:
    def __init__(self,gelis_zamani,tir_plakasi, ulke, ton_20_adet, ton_30_adet, yuk_miktari,maliyet):
        self.gelis_zamani = gelis_zamani
        self.tir_plakasi = tir_plakasi[-3:]
        self.ulke = ulke
        self.ton_20_adet = int(ton_20_adet)
        self.ton_30_adet = int(ton_30_adet)
        self.yuk_miktari = int(yuk_miktari)
        self.maliyet = maliyet
```

Bu satır, Tir adında bir sınıf (class) başlatır. Bu sınıf, olaylar. csv dosyasındaki tır verilerini temsil eder.
Bu sınıfın başlatıcı metodu (__init__), bir TIR nesnesi oluşturulduğunda çağrılır ve tırın özelliklerini belirler. Bu özellikler şu şekildedir: gelis_zamani, tir_plakasi, ulke, ton_20_adet, ton_30_adet, yuk_miktari, maliyet.
Bu özellikler, tır nesnesinin sahip olacağı temel bilgileri içerir ve bu bilgilerin sınıf içinde tutulmasını sağlar. Bu sayede, bir tır nesnesi oluşturulduğunda, bu özelliklere erişebilir ve bu bilgiler üzerinde işlemler yapabilirsiniz.

-Gemi

```
class Gemi:
    def __init__(self, gelis_zamani, gemi_adi, kapasite,gidecek_ulke):  # gemiler.csv parametreleri
        self.gelis_zamani = gelis_zamani
        self.gemi_adi = gemi_adi
        self.yuk_durumu = 0  # Kapasite dolumunu kontrol etmek amacıyla başlangıç değeri belirledik.
        self.kapasite = int(kapasite)
        self.gidecek_ulke = gidecek_ulke
```

Bu satır, Gemi adında bir sınıf başlatır. Bu sınıf, gemiler. csv dosyasındaki gemi verilerini temsil eder. Aynı tır sınıfındaki gibi belirli parametreleri, özellikleri vardır ve üzerinden işlemler yapılır.

-Liman

```
class LimanOtomasyonu:
    def __init__(self, tir, gemi):
        self.tirlar = self.tir_degerleri(tir)
        self.gemiler = self.gemi_degerleri(gemi)
        self.yukle()
```

Bu satır, LimanOtomasyonu adında bir sınıf başlatır. Bu sınıf, liman operasyonlarını yöneten temel bir sınıftır. Bu sınıf, limandaki yük indirme-yükleme otomasyon sistemini simüle etmek için kullanılır. İlk başta, sınıfın kurucu metodunda, sınıfın başlatılması için iki önemli işlem gerçekleştirilir.

Tırlar ve gemiler adında iki özellik oluşturulur ve bunlar, sınıfın başlatılma aşamasında parametre olarak verilen tır ve gemi verilerini içerir. self. tir_degerleri ve self. gemi_degerleri metotları bu verileri sırasıyla okuyarak ilgili sınıfların nesnelerini oluşturur.

2. CSV DOSYASI KULLANIMI

```
def tir_degerleri(self, dosya):
    tirlar = []
    with open(dosya, 'r', newline='', encoding='ISO-8859-9') as file:
        reader = csv.reader(file)
        next(reader)  # ilk satırda verilerin isimleri yazdığı için burayı atlamak amaçlı
        for row in reader:
            tirlar.append(Tir(row[0], row[1], row[2],row[3],row[4],row[5],row[6]))
    return sorted(tirlar, key=lambda x: x.tir_plakasi)

def gemi_degerleri(self, dosya):
    gemiler = []
    with open(dosya, 'r', newline='', encoding='ISO-8859-9') as file:
        reader = csv.reader(file)
        next(reader)  # ilk satırda verilerin isimleri yazdığı için burayı atlamak amaçlı
        for row in reader:
            gemiler.append(Gemi(row[0], row[1],row[2],row[3]))
    return sorted(gemiler, key=lambda x: x.gemi_adi)
```

Bu iki metot, LimanOtomasyonu sınıfının özelliklerini (tırlar ve gemiler) başlatmak için kullanılır. İki metot da bir CSV dosyasından veri okur ve bu verileri uygun sınıfların nesnelerine dönüştürerek bir listeye ekler. Ardından, bu listeleri sıralayarak ilgili sınıfların nesneleriyle döndürür. With open() ile dosyayı açar ve reader özelliği ile dosyayı okur. Okunan değerler tırlar boş listesine buradan kullanılmak üzere eklenir.

3. YÜKLEME-İNDİRME İŞLEMLERİ FONKSİYONU

```
def yukle(self):
```

Yükle fonskiyonu oluşturarak yükleme-indirme işlemlerini başlatmış oluyoruz.

4. TIRLARDAKİ TON ADET SAYILARINA GÖRE İSTİF ALANLARINA YÜK BIRAKMA DURUMU

3 adet durum bulunmakta ilki istif alanlarına yük aktarımıdır.

Bunun için aşağıdaki kodu yazmış bulunmaktayız.

```
istif_alani1 = []
istif_alani2 = []
istif_kapasitesi = 750
for tir in self.tirlar:
    if tir.ton_20_adet == 1:  # Ton adetinin 1 ve 0 olma durumuna göre kaç ton yük yükleneceğini çeken koşullandırmalar
        if sum(istif_alani1) + 20 <= istif_kapasitesi:
            istif_alani1.append(20)
        elif sum(istif_alani2) + 20 <= istif_kapasitesi:
            istif_alani2.append(20)
        else:
            print("İstif alanları dolmuştur!")
            break
    if tir.ton_30_adet == 1:
        if sum(istif_alani1) + 30 <= istif_kapasitesi:
            istif_alani1.append(30)
        elif sum(istif_alani2) + 30 <= istif_kapasitesi:
            istif_alani2.append(30)
        else:
            print("İstif alanları dolmuştur!")
            break
```

istif_alani1 ve istif_alani2 adında iki boş liste oluşturulur. Bu listeler, yüklerin istiflenmesi için kullanılacak iki farklı istifleme alanını temsil eder. İstif_kapasitesi adında bir sabit oluşturulur.(750)

Daha sonrasında ton_adet değeri kontrol edilir. Ton_20_adet değeri 1 olma durumunda yük 20 ton olacak ve önce 1.istif alanına daha sonra 2.istif alanına eklenecektir. Eklenmeler koşulları sağlamadığında dolduğu anlamına gelecek ve ekrana çıktı olarak bildirim gelecektir. Aynı işlemi hem 20 hem 30 ton için gerçekleştiririz. Bu işlem kodlarının hizasındaki bitime de break fonksiyonu ekleyerek sonsuz döngüyü ve karışıklığı engelleriz.

5. TIRLARDAN GEMİLERE YÜKLEME DURUMU

```
                                # TIRLARDAN GEMİLERE YÜKLEME DURUMU
for tir in self.tirlar:
    for gemi in self.gemiler:
        if tir.ulke == gemi.gidecek_ulke:  # Ülkeleri eşleşen tır ve gemiler yüklenir
            print(f"Tır {tir.tir_plakasi} yük indiriyor...")
            if gemi.yuk_durumu + tir.yuk_miktari > 0.95 * gemi.kapasite:  # Yük geminin kapasitesini aşma durumu
                dolma_durumu = int(0.95 * gemi.kapasite) - gemi.yuk_durumu  # 95'ini hesaplar
                if dolma_durumu > 0:  # Miktar yüzde 95'inden fazla ise gemi ayrılır.
                    tir.yuk_miktari = tir.yuk_miktari - dolma_durumu
                    gemi.yuk_durumu = gemi.yuk_durumu + dolma_durumu
                    print(f"{gemi.gemi_adi}.sıra numaralı gemi kapasitesini doldurdu.Limandan ayrılıyor. Yük durumu: {gemi.yuk_durumu}
            else:
                gemi.yuk_durumu += tir.yuk_miktari  # Tırda kalan yüklerin aktarılma durumu
                print(f"{gemi.gemi_adi}.sıra numaralı gemi yük durumu: {gemi.yuk_durumu}")
                break  # eğer tırda yük bittiyse break ile yük alma bitirilir.
```

Bu kod bloğunda tırlardaki yükleri plaka numarasına göre gemilere sıra numarasına göre yükleme işlemini gerçekleştirir.

Tirlar ve gemiler listelerini kullanarak verilere erişiriz.

if tir.ulke == gemi.gidecek_ulke: koşulu, tırın bulunduğu ülke ile geminin gideceği ülkenin eşleşip eşleşmediğini kontrol eder. Eğer eşleşiyorsa, tırdan bu gemiye yük yüklenecek demektir.

print(f"Tır {tir. tir_plakasi} yük indiriyor..."): Eşleşme durumunda, ilgili tırın plakası ekrana yazdırılır.

Geminin yük almayı bitirmesi için %95'inin dolması gerekmektedir. Bunun için if-else yapısını kullandık.
gemi.kapasite değeri, geminin yük durumunun, tırdan gelecek yükü alırsa geminin kapasitesinin %95'ini aşıp aşmadığını kontrol eder.
Eğer aşarsa, bir dolma_durumu değişkeni hesaplanır
dolma_durumu = int(0.95 * gemi.kapasite) - gemi.yuk_durumu. Bu, geminin kalan kapasitesinin %95'ini hesaplar ve geminin şu anki yük durumundan çıkararak geriye kalan dolma durumunu belirler.

Gemide kalan kapasite yoksa aşağıdaki işlemleri gerçekleştirir.
-Tır yük miktarından gemiye aktarılan dolma durumu çıkarılır.
-Gemideki yük durumuna dolma durumu eklenir.
-Ekrana gemi kapasitesinin dolduğuna dair bir mesaj yazdırılır

Gemide kalan kapasite varsa işlem devam eder.

6. HER GEMİ VE TIR İÇİN SÖZLÜK DEĞİKENİ OLUŞTURMA VE VERİLERE ULAŞMA ÖRNEKLERİ
   Bu kısımda tır ve gemi bilgilerini sözlük (dictionary) formatında saklamak ve kullanıcıdan alınan bilgiye göre ilgili aracın bilgilerini ekrana yazdırmak için bir mekanizmayı içerir.

```
tir_sozluk = {tir.tir_plakasi: {"Geliş zamanı":tir.gelis_zamani,"Ülke": tir.ulke,"Yük miktarı": tir.yuk_miktari,"20 tonluk konteynır adeti":
gemi_sozluk = {gemi.gemi_adi: {"Geliş zamanı":gemi.gelis_zamani,"Gideceği Ülke": gemi.gidecek_ulke,"Yük durumu": gemi.yuk_durumu,"Kapasite":
# Kullanım
bilgi = input("Hangi aracın bilgisine erişmek istersiniz? (TIR,GEMI)")
if bilgi == "TIR":
    plaka = input("Tır plakası giriniz:")
    if plaka in tir_sozluk:
        print(f"Tır Bilgileri: {tir_sozluk[plaka]}")
    else:
        print(f"{plaka} plakalı tır bulunamadı.")
elif bilgi == "GEMI":
    gemi_adi = input("Geminin adını giriniz:")
    if gemi_adi in gemi_sozluk:
        print(f"Gemi Bilgileri: {gemi_sozluk[gemi_adi]}")
    else:
        print(f"{gemi_adi} sıralı gemi bulunamadı.")
else:
    print("Düzgün bir ifade girin.")
```

   tir_sozluk ve gemi_sozluk adında iki sözlük oluşturulur.
   tir_sozluk: tırların plakalarını anahtar olarak kullanarak, her bir tırın bilgilerini içeren bir alt sözlük oluşturur.
   gemi_sozluk: Gemilerin adını anahtar olarak kullanarak, her bir geminin bilgilerini içeren bir alt sözlük oluşturur.
   Bu alt sözlüklerde tır ve gemiye ait diğer özellikleri de içerir. Böylece tüm verilere erişmiş oluruz.
   Bu erişimi sağlamak için de kullanıcı input ile bir if-elif-else yapısında seçim yapar.
   Seçimine göre ekranda bilgiler çıktılanır.

7. Çalıştırma

```
tir_dosya = 'olaylar.csv'
gemi_dosya = 'gemiler.csv'
calistirici = LimanOtomasyonu(tir_dosya, gemi_dosya)
```

   Bu kod, LimanOtomasyonu sınıfını kullanarak bir liman otomasyonu uygulamasını başlatır. Ödevde belirtilmiş olan csv dosyalarını değişken olarak verdikten sonra otomasyona bu dosyalar girilir.
   Otomasyon sınıfına calistirici adında değişken atayarak sınıf nesnesi oluşturmuş oluyoruz. Böylece işlemler çalışmaya başlamış olmaktadır.

4.SONUÇ

Bu proje kapsamında gerçekleştirilen çalışmalar sonucunda, tırların plaka numaralarına göre sıralı olarak yük indirmesi ve gemilerin numaralarına göre sıralı olarak yük yüklemesi, sistemin düzenli bir şekilde işlemesini sağlamıştır. "olaylar.csv" ve "gemiler.csv" dosyaları aracılığıyla gerçekleştirilen veri girişi ile sistem dinamik ve değişken senaryolara uyum sağlamıştır. Vinç işlemleri, yükleri istif alanlarına yerleştirme ve gemilere yüklemesi İstif alanlarının kapasite kontrolü ve doluluk durumu da takip edilmiştir.

Sonuç olarak, bu Python programı, liman otomasyon işlemlerini gerçekleştirmek için kullanışlı bir araçtır. Kullanıcıların ihtiyaçlarına yönelik olarak çeşitli işlemler yapabilir ve sonuçları kolayca elde edebilir.

Bu projenin başarıları ve karşılaşılan zorluklar, gelecekte benzer simülasyon sistemlerinin geliştirilmesi veya liman operasyonlarının daha etkin bir şekilde planlanması için değerli bir temel oluşturmuştur. Projenin geliştirmelerle daha da güçlendirilmesi, gerçek hayatta karşılaşılabilecek senaryolara daha iyi adapte olmasını sağlayabilir.

Ayrıca, bu projenin grup çalışmasıyla tamamlanması, bizlere takım çalışmasının gücünü ve zorluklarını gösterdi. Grup içi işbirliği sayesinde farklı becerilere ve bakış açılarına sahip arkadaşlarımızın fikirleri, projenin daha iyi bir şekilde geliştirilmesine katkı sağladı. Aynı zamanda fikir ayrılıklarından kaynaklanan zorluklarla da karşılaştık. Bu deneyimler, yazılım geliştirme sürecinde işbirliği ve iletişim becerilerimizi geliştirmemize yardımcı oldu.

5.KAYNAKÇA

https://forum.yazbel.com/t/csv-dosyasi-okuma/7001
https://muhammeddincer.com/python-ile-csv-dosya-okuma-ve-yazma/