



T.C

**KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ MÜHENDİSLİK
VE DOęA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİęİ PROGRAMI**

LİMAN OTOMASYONU

Hazırlayanlar:

Hilal AYDIN 220501030

Zehra KANDAZ 220501026

DERS SORUMLUSU:

PROF. DR. TARIK DURU

13.12.2023

İÇİNDEKİLER

1. ÖZET

2. GİRİŞ

3. KOD İÇERİĞİ

3.1 Tır Sınıfı

3.2 Gemi Sınıfı

3.3 Dosyaların okunması

3.4 Tır Bilgileri Fonksiyonu

3.5 Gemi Bilgileri Fonksiyonu

3.6 Liman Simülasyonu Fonksiyonu

3.7 Arayüz

3.8 Time Kütüphanesi:

3.9 Tkinter Kütüphanesi

3.10 Pandas Kütüphanesi:

4. SONUÇ VE ÖĞRENİLEN DERSLER

5.KAYNAKÇA

6.GİTHUB BAĞLANTILARI

1.ÖZET:

Python dilini kullanarak yapmış olduğumuz bu liman otomasyonu projesinde bir limana belirli zamanda gelen tırların yüklerini istif alanlarına indirilmesi sağlanır. Ardından yine belirli zamanlarda bu limana gelen dört farklı ülkeye gidecek olan gemilere kendi kapasitesini aşmayacak şekilde gidecekleri ülkeye göre yirmi ve otuz tonluk yüklerle yüklenir. Kapasitesinin yüzde doksan beşi dolan gemilerin limandan ayrılması sağlanır. Tüm bunları gösteren otomasyon sisteminin simülasyonunu arayüzle birlikte oluşturulur.

2.GİRİŞ



Yukarıda bulunan liman otomasyonunun arayüzünde ilgili kısma simülasyonun ne zamana kadar çalıştırılmak istendiği girilir ardından "simülasyonu başlat" butonuna tıklanarak çalıştırılır.

Her t anında gerçekleşen olaylar "simülasyon" bölümünde görünür.

TIR'lardan indirilen yükler "istif alanı" bölümüne eklenir.

Gemilere yüklenen yükler istif alanından çıkarılır.

TIR'ın plakası girildikten sonra "TIR Bilgilerini Göster" butonuna tıklandığında girilen TIR'ın yük bilgileri "TIR Bilgileri" bölümünde gözükür.

Geminin ismi girilip ardından "Gemi Bilgilerini Göster" butonuna tıklandığında girilen geminin yük bilgileri "Gemi Bilgileri" bölümünde gözükür.

"Çıkış" butonuna tıklandığında ise programdan çıkılır.

3. KOD İÇERİĞİ:

3.1 Tır Sınıfı:

- TIR'lar için bir TIR sınıfı oluşturulur. Bu sınıf 3 özellik içerir: geliş zamanı, tır plakası ve yük bilgisi.
- Yük bilgisi sözlük türündedir ve şunları içerir: ülke, 20 ton adet, 30 ton adet, yük miktarı ve maliyet.

3.2 Gemi Sınıfı:

- Gemiler için bir gemi sınıfı oluşturulur. Bu sınıf 4 özellik içerir: geliş zamanı, gemi adı, kapasite ve yük bilgisi.
- Yük bilgisi sözlük türündedir ve şunları içerir: ülke, 20 ton adet, 30 ton adet, yük miktarı ve maliyet.
- 20 ton adet, 30 ton adet, yük miktarı ve maliyet parametrelerine default olarak sıfır atanır.

3.3 Dosyaların okunması:

- Pandas kütüphanesinin read_csv() metodu ile "olaylar.csv" dosyası okunur.
 - Bir DataFrame oluşur.
 - iterrows() metodu ile DataFrame'nin satırları gezilir.
 - Her satır için bir TIR nesnesi oluşturulur ve bu nesneler "tirlar" adlı listeye eklenir.
 - Bu liste TIR'ların geliş zamanı ve plakalarına göre küçükten büyüğe sıralanarak sirali_tirlar adında yeni bir liste oluşturulur.
-
- read_csv() metodu ile "gemiler.csv" dosyası okunur.
 - Bir DataFrame oluşur.
 - iterrows() metodu ile DataFrame'nin satırları gezilir.

- Her satır için bir gemi nesnesi oluşturulur ve bu nesneler "gemiler" adlı listeye eklenir.

3.4 Tır Bilgileri Fonksiyonu:

- TIR bilgilerine ulaşmak için bir fonksiyon oluşturulur.
- Bu fonksiyonda kullanıcıdan TIR plakası alınır.
- sirali_tirlar listesinde gezilir.
- Her bir TIR plakasının kullanıcının girdiği plakayla aynı olup olmadığı kontrol edilir. Eğer aynıysa bu TIR'ın bilgileri her bir geliş zamanı için 1,5 saniye arayla ekrana basılır.

3.5 Gemi Bilgileri Fonksiyonu:

- Gemi bilgilerine ulaşmak için bir fonksiyon oluşturulur.
- Bu fonksiyonda kullanıcıdan gemi adı alınır.
- Gemiler listesinde gezilir.
- Her bir gemi adının kullanıcının girdiği ad ile aynı olup olmadığı kontrol edilir. Eğer aynıysa bu geminin bilgileri ekrana basılır.

3.6 Liman Simülasyonu Fonksiyonu:

- Simülasyonu çalıştırmak için bir fonksiyon oluşturulur.
- Bu fonksiyon 1. zamandan kullanıcıdan alınan zamana kadar çalışır.
- Her t zamanı için "sirali_tirlar" listesindeki TIR'lar gezilerek bu TIR'ların geliş zamanı ile zamanın eşit olup olmadığı kontrol edilir.
- Eşit olan TIR'ların yükleri istif alanına yüklenir.
- t zamanında gelen her TIR için ekrana "TIR limana geldi" yazısı basılır. TIR'ın bilgisi basılır. Yükünü indiren her TIR için "TIR yükünü indirdi" yazısı basılır.
- İstif alanında gerçekleşen tüm yük indirme bindirme işlemleri de ekranda gözüktür.
- Yüklerini indiren TIR'ların yük bilgileri sıfırlanır.
- Her t zamanı için "gemiler" listesinde gezilerek bu gemilerin geliş zamanının zamana eşit olup olmadığı kontrol edilir.
- Eşit olan gemiler "limandaki_gemiler" listesine eklenir.
- t zamanında gelen her gemi için ekrana "Gemi limana geldi" yazısı basılır. Geminin

bilgisi basılır.

- İstif alanındaki yüklerin ülkeleri ile gemilerin ülkelerinin aynı olup olmadığı kontrol edilir. Aynı olan yükler eğer geminin kapasitesi yetiyorsa gemiye yüklenir.
- Yüklenen her gemi için "Yük gemiye yüklendi" yazısı basılır.
- Yüklenen gemilerin yük bilgisi güncellenir.
- Yüklenen yükler istif alanından çıkarılır.
- Kapasitesinin %95'i dolan gemi limandan ayrılır ve "Gemi limandan ayrıldı" yazısı basılır.
- İstif alanı boşsa "istif alanı boş" yazısı ekrana bastırılır.

3.7 Arayüz:

- Arayüz için bir pencere oluşturulur.
- title() metodu ile pencerenin başlığı "Liman Otomasyonu" olarak belirlenir.
- config(bg="") metodu ile pencerenin arka plan rengi değiştirilir.
- Frame() metodu ile bir çerçeve oluşturulur.
- Label() metodu ile bu çerçevenin içine 4 tane etiket yerleştirilir.
- Her bir etiket bölmesi için birer başlık etiketi oluşturulur. Bunlar: Simülasyon, TIR Bilgileri, Gemi Bilgileri ve İstif Alanıdır.
- Grid(row=, column=, padx= pady=) metodu ile her bir etiket çerçeveye yerleştirilir. Row ve column parametreleri ile hangi konumlarda bulunacakları belirlenir. Padx ve pady parametreleri ile x eksenine ve y eksenine uzaklıkları belirlenir.
- Entry() metodu ile kullanıcıdan giriş alınacak bölmeler oluşur.
- Button() metodu ile 4 tane buton oluşturulur. Bunlar: Simülasyonu Başlat, Çıkış, TIR Bilgileri Göster ve Gemi Bilgilerini Göster.
- Bu widget'larda bulunan text parametresi üzerinde yazacak yazıyı, width genişliğini, height yüksekliğini, bg arka plan rengini, fg yazı rengini, anchor widget'ın konumunu belirler.

3.8 Time Kütüphanesi:

- "time" kütüphanesi zamanla ilgili işlemleri gerçekleştirmek için kullanılır. Kodda kullanılan time.sleep() fonksiyonu, programın belirli bir süre boyunca duraklamasını sağlar.

3.9 Tkinter Kütüphanesi:

- "tkinter" kütüphanesi, Python için standart GUI kütüphanesidir.
- Tk() ile pencere oluşturulur.
- title() ile pencerenin başlığı belirlenir.
- config() ile bir widget'ın özellikleri değiştirilir. Bu özelliklerden bazıları: metin, yükseklik, genişlik, yazı rengi, arkaplan rengi.
- Frame() ile bir çerçeve oluşturulur ve bu çerçeve içine başka widget'lar eklenir.
- Label() ile metin veya resim içeren etiketler oluşturulur.
- Button() ile buton oluşturulur. Butona tıklandığında gerçekleşecek işlemler için birfonksiyon belirtilir.
- Entry() ile kullanıcıdan metin girişi almak için bir giriş kutusu oluşturulur.
- pack() ile widget'lar pencereye yerleştirilir.
- grid() ile widget'lar pencereye, row ve column parametreleri belirtilerek belirli satır ve sütunlara yerleştirilir.
- mainloop() ile tkinter uygulamasının ana döngüsü başlatılır.

3.10 Pandas Kütüphanesi:

- "pandas" veri işlemesi ve analizi için kullanılan bir kütüphanedir.

4.SONUÇ VE ÖĞRENİLEN DERSLER:

Visual Studio Code üzerinde Python kullanarak geliştirdiğimiz Liman Otomasyonu projesi, Pandas kütüphanesinin sunduğu avantajları kullanarak CSV dosyaları üzerinde daha verimli bir şekilde çalışabilmemizi sağladı. Bu kütüphane sayesinde veri analizi ve manipülasyonu konusunda daha etkili bir şekilde çalışabildik.

Pandas kütüphanesinin yanı sıra, arayüz oluşturmak için kullandığımız Tkinter kütüphanesi ve programın çalışma süresini de kontrol edebildiğimiz Time kütüphanesini de programda kullanarak daha iyi öğrenme fırsatımız oldu.

Ayrıca bu projeyi yaparken Python'da nesneye yönelik programla kavramlarından biri olan sınıflarla çalışmayı da uygulayarak daha iyi kavramış olduk. Sınıfların kullanımı, projenin daha düzenli ve yönetilebilir olmasını sağladı

5.KAYNAKÇA:

- <https://docs.python.org/tr/3/library/time.html>
- <https://medium.com/@ilyaskaraca/python-gui-programlama-tkinter-d63a99b43179>
- <https://chat.openai.com/>

6.GİTHUB BAĞLANTILARI:

Hilal Aydın:

<https://github.com/HilalAydin>

Zehra Kandaz:

<https://github.com/ZehraKandaz>