

TC
KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ LİSANSÜSTÜ
EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

LİMAN OTOMASYONU

Hazırlayan
HATİCE REYHAN ÇALIŞKAN 220501001
BETÜL CANOL 220501009

Ders Sorumlusu
Prof. Dr. HÜSEYİN TARIK DURU

GitHub Linkleri:
<https://github.com/ReyhanClskn/Port-Automation>
<https://github.com/BetulCnl/port-automation>

Tarih
16/10/2023

1.ÖZET

Bu ödevde, bir limandaki yük indirme-yükleme otomasyon sisteminin simülasyonunun Python kullanılarak yapılması istenmektedir. Senaryo şöyledir: yüklerini istif alanına indirecek tırlar ve istif alanından indirilen yükleri alıp istenilen ülkeye götürecek olan gemiler bulunmaktadır.



Görünüm en son bu şekildedir.

2.GİRİŞ

Bu senaryoda bulunan tırlar ve gemiler bizde bir csv dosyası olarak bulunuyor. Dosyamızda tırların geliş zamanı, plakası, geldikleri ülke, taşıdıkları yirmişer ve otuzar yük sayısı, toplam yük miktarı bulunmaktadır.

Gemiler csv dosyamızda da geminin geliş zamanı, geminin adı, taşıyabileceği yük kapasitesi ve yükleri ulaştıracağı ülke bulunmakta.

İşleyiş şu şekilde: öncelikle tırlar csv dosyamızda yazan zaman sıralamasına göre limana geliyorlar. Tırlar geldikçe sırasıyla yüklerini bırakıyorlar eğer aynı anda gelirlerse plaka numaraları küçükten büyüğe olacak şekilde yüklerini istif alanına bırakıyorlar. İki istif alanımız bulunmakta, Bir istif alanının kapasitesi 750 tondur. Gemilerden öncelikli olarak ilk gelen ve eğer aynı anda birden fazla gemi gelirse ismi küçük olan ilk yüklenecek şekilde yükleri yüklenirler ve istenilen ülkeye yükleri götürürler.

Csv dosyamız programda şu şekilde gözükmemektedir;

```
geliş_zamanı,tır_plakası,ülke,20_ton_adet,30_ton_adet,yük_miktarı,maliyet
2,41_kostu_046,Oceania,0,1,30,30000
6,41_kostu_616,Neverland,1,0,20,20000
6,41_kostu_616,Neverland,1,0,20,20000
6,41_kostu_998,Neverland,1,0,20,20000
6,41_kostu_751,Lilliputa,0,1,30,30000
6,41_kostu_579,Mordor,0,1,30,30000
7,41_kostu_978,Lilliputa,0,1,30,30000
7,41_kostu_487,Lilliputa,1,0,20,20000
7,41_kostu_039,Lilliputa,0,1,30,30000
7,41_kostu_201,Mordor,0,1,30,30000
8,41_kostu_338,Mordor,1,0,20,20000
8,41_kostu_923,Lilliputa,1,0,20,20000
8,41_kostu_053,Mordor,1,0,20,20000
8,41_kostu_748,Mordor,1,0,20,20000
8,41_kostu_983,Lilliputa,1,0,20,20000
8,41_kostu_286,Oceania,0,1,30,30000
8,41_kostu_540,Lilliputa,1,0,20,20000
13,41_kostu_015,Neverland,0,1,30,30000
13,41_kostu_622,Oceania,1,0,20,20000
13,41_kostu_087,Oceania,1,0,20,20000
13,41_kostu_022,Oceania,0,1,30,30000
13,41_kostu_241,Mordor,1,0,20,20000
13,41_kostu_420,Lilliputa,0,1,30,30000
15,41_kostu_637,Lilliputa,1,0,20,20000
15,41_kostu_982,Mordor,1,0,20,20000
```

3.YÖNTEM

3.1

Projemizde ilk olarak tır ve gemi sınıflarını oluşturduk. Tırların plakaları ve kaçar ton yükten kaç adet taşıdıkları bilgisi tır sınıfı sayesinde elimizde bulunuyor. Gemilerin plakası hangi ülkeye gidecekleri ve taşıdıkları yük miktarı da gemi sınıfımızda bulunuyor. Bu bilgileri sınıflarda bulundurup sonra ayrı olarak asıl geçiş işlemlerinin gerçekleşeceği liman sınıfında bu bilgileri kullanarak otomasyonu gerçekleştiriyoruz.

3.2

Liman sınıfında otomasyon çalışırken her seferinde istif alanının dolu olup olmadığını kontrol eden bir fonksiyonumuz bulunmakta bu şekilde tırların ne zaman yükleri indirip gemilerin ne zaman yükleneceğini kontrol ediyoruz. Tırların yükleri iki koşula göre indirdiğinden emin oluyoruz; ilk olarak geliş zamanına göre, önce gelen tır yükünü önce bırakır ancak aynanda gelen tırlar mevcutsa bu durumda plakası küçük olan tır yükünü önce indirir daha sonrasında onunla birlikte gelmiş olan tır plakasının kendisinden sonra gelen tırdan küçük veya büyük olması fark etmeksizin yükünü bırakır. Gemilerin yükleri yükleniş şekli de bu mantıkta ilerler. Gemi yüklenmesi gereken miktar kadar yükü dolu istif alanından çekip alır ve csv dosyasında görünen götürmesi istenilen ülkeye götürdüğü kullanıcıya çıktı olarak başarılı bir şekilde iletildiği hakkında çıktı verir. Bunların yanında iki fonksiyon ile kullanıcı plakasını girdiği tır veya geminin ne kadar yük yüklendiği ve nereye gittiğini de görebilmektedir. En sonda bulunan kod da verilen çıktıların arayüz görünümünde çıkmasını sağlar.

4. SONUÇ VE ÖĞRENİLEN DERSLER

Sınıf oluşturma'nın önemi ve birden fazla sınıf oluşturup birbirleriyle ilişkilerini daha derin kavramış olduk. Sınıfların kullanılarak yazılan ve kullanmadan yazılmaya çalışan kodu da bu ödev sayesinde denedik. Birbirleri arasında nasıl ve neden bağlantılar kurmamız gerektiğini nasıl kuracağımızı gözlemledik. Bu ödevin gerçek hayatta karşımıza çıkan problemleri nasıl çözümleyeceğimizi de gösterdiğini düşünüyoruz. Kodu yazarken fonksiyonların nasıl çalıştığından çok hangi sırayla ve nasıl bir döngü içerisinde çalışması gerektiği konusunda kafamız karıştı. Bu da gerçek hayatta bulunup onları çözmek için yazacağımız kodlarda nasıl bir düşünce aşaması geçirmemiz gerektiğini göstermiş oldu. Kod olarak yazmaya başlamadan önce çok kez kafamızda otomasyon işleyişinin nasıl olacağını düşünüp tasarladık.

Ayrıca kodu yazarken sürekli kaydetmenin çok önemli olduğunu bildiğimiz gibi önemli değişikliklerde kaydettiğimiz an o anki kaydı isimlendirmenin de çok önemli olduğunu gözlemlemiş olduk. Kodda sürekli değişiklikler ve oynamalar yaptığımız için bir kısmı düzelirken diğer kısmı bozulabiliyordu ve o bozulan kısmın doğru çalıştığı hali hangi kaydedilen kısımda olduğunu bulmanın çok zor olduğuyla karşılaşmış bulunduk. Sonrasında önemli değişiklikler yapmadan önce her kaydedişimizde aşamaları isimlendirdik ve böylece büyük karışıklıkların sonradan da olsa önüne geçebilmeyi öğrendik.

Yeni olarak bir de threading modülünü öğrendik. Bu kodda bu modülü, farklı iş parçacıklarının aynı anda çalışmasını sağlamak için kullandık. Programın performansını arttırmak için kullandık.

5. KAYNAKÇA

<https://youtu.be/YXPyB4XeYLA?si=t99ztAsf7Fw3M9DR>

<https://mcsp.wartburg.edu/zelle/python/ppics3/>

<https://www.geeksforgeeks.org/python-tkinter-tutorial/>

https://www.w3schools.com/python/python_classes.asp

<https://docs.python.org/3/>