**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**Python Dili ile Liman Otomasyonu**

**Hazırlayan**

**Yavuz Selim Gürsoy Talha Tuna**

220501003 220502015

**DERS SORUMLUSU**

**ARŞ. GÖR. ERAY DURSUN**

**13 ARALIK 2023**

**İÇİNDEKİLER**

[1. ÖZET 3](#_Toc153364062)

[2. GİRİŞ 3](#_Toc153364063)

[3. YÖNTEM 3](#_Toc153364064)

[3.1 Truck sınıfı 3](#_Toc153364065)

[3.2 Ship sınıfı 4](#_Toc153364066)

[3.3 CargoBay sınıfı 4](#_Toc153364067)

[3.4 Group() fonksiyonu: 5](#_Toc153364068)

[3.5 Order() fonksiyonu: 5](#_Toc153364069)

3.6 Simulation() fonksiyonu:

[3.7 Main() fonksiyonu: 5](#_Toc153364070)

[4. SONUÇ VE ÖĞRENİLEN DERSLER 6](#_Toc153364071)

[5. KAYNAKÇA 6](#_Toc153364072)

6. GITHUB BAĞLANTILARI

# 

# ÖZET

Bu rapor, Python dilinde oluşturulmuş ve bir limandaki yük indirme-yükleme otomasyon sisteminin simülasyonunu gerçekleştirir. İlgili kodlar fonksiyonlar halinde ayrılmış ve her bir fonksiyonun işlevi, kullanımı ve iç çalışma mantığı ayrıntılı olarak açıklanmıştır.

# GİRİŞ

Bu Python projesi, lojistik süreçleri simüle etmek amacıyla geliştirilmiştir. Python, projede kullanılan güçlü ve esnek bir programlama dilidir, bu sayede karmaşık lojistik senaryolarını kolayca modelleme ve işleme imkânı sağlar. Projede, kamyonların taşıma işlemleri ve gemilerin yük boşaltma işlemleri gibi lojistik etkileşimlere yer verilmiştir. Bu rapor, projede kullanılan sınıflar, fonksiyonlar ve temel mantığı detaylı bir şekilde açıklamaktadır.

# YÖNTEM

Bu bölüm, Python programının temel yapı taşlarını içerirken, aynı zamanda lojistik süreçlerin simülasyonu için nasıl adapte edildiğini ve hangi yöntemlerin kullanıldığını anlamak için okuyuculara kapsamlı bir rehber sunmaktadır. Her bir bileşenin işlevi, çalışma prensibi ve kullanılan Python özellikleri bu bölümün hemen devamında detaylı olarak incelenecektir.

## Truck sınıfı

Bu sınıf bir tır sınıfını belirtir. Bu sınıf, kamyonların özelliklerini ve bu özelliklere erişim sağlayan bir metodunu içerir. İşlevleri ve fonksiyonları şu şekildedir:

**def\_ \_init\_ \_(self, arriveTime, licensePlate, destination, ton20, ton30, truckLoad, cost):** Bu sınıfın başlatıcı metodu (constructor) olarak kullanılır. Kamyonun özelliklerini alarak bir kamyon örneği oluşturulur. Kamyonun varış zamanı (arriveTime), plakası (licensePlate), varış noktası (destination), taşıma kapasitesi (truckLoad), maliyeti (cost), 20 ton yük miktarı (ton20) ve 30 ton yük miktarı (ton30) gibi özellikleri bulunmaktadır. Bu özelliklerin bir kısmı kullanım kolaylığı olması açısından tamsayı türüne çevrilir.

**def printInfo(self):** Bu üye fonksiyon, kamyonun bilgilerini renkli bir şekilde ekrana yazdırmak için kullanılır. Kamyonun plakası, varış zamanı, varış noktası, taşıma kapasitesi, maliyeti, 20 ton yük miktarı ve 30 ton yük miktarı gibi bilgileri ekrana yazdırır. Çıktı aşamasında bu fonksiyon kullanılmaz. Sadece kod yazarken programcıya ek bir görüş sağlamak amacıyla yazılmıştır.

## Ship sınıfı

Bu sınıf gemi sınıfını tanımlar. Bu sınıf, gemilerin özelliklerini ve bu özelliklere erişim sağlayan bir metodunu içerir. İşlevleri ve fonksiyonları şu şekildedir:

**def\_ \_init\_ \_(self, arriveTime, name, capacity, destination):** Bu sınıfın başlatıcı metodu (constructor) olarak kullanılır. Geminin özelliklerini alarak bir gemi örneği oluşturulur. Geminin varış zamanı (arriveTime), ismi (name), taşıma kapasitesi (capacity), varış noktası (destination) gibi özellikleri bulunmaktadır. Ayrıca, geminin minimum kapasitesini (min) ve sabit kapasitesini (constcap) hesaplar. Bu özelliklerin bir kısmı kullanım kolaylığı olması açısından tamsayı türüne çevrilir.

**def printInfo(self):** Bu metod, geminin bilgilerini renkli bir şekilde ekrana yazdırmak için kullanılır. Geminin ismi, varış zamanı, varış noktası ve taşıma kapasitesi gibi bilgileri ekrana yazdırır. Çıktı aşamasında bu fonksiyon kullanılmaz. Sadece kod yazarken programcıya ek bir görüş sağlamak amacıyla yazılmıştır.

**def isFull(self):** Bu metod, geminin dolup dolmadığını kontrol eder. Eğer geminin kapasitesi 0 ise veya sabit kapasitesinden mevcut kapasitesini çıkardığında minimum kapasitenin %95'inden büyük veya eşitse, gemi dolmuş kabul edilir ve True döndürülür. Aksi durumda False döndürülür. Bu üye fonksiyon, geminin taşıma kapasitesine göre yüklenip yüklenmediğini kontrol eder.

## CargoBay sınıfı

Bu sınıf yük alanı sınıfını tanımlar. Bu sınıf, bir geminin yük alabileceği bir alanı temsil eder. Sınıfın işlevleri ve fonksiyonları şu şekildedir:

**def\_\_init\_\_(self, bayName):** Sınıfın başlatıcı metodu (constructor) olarak kullanılır. Yük alanının ismi (bayName), kapasitesi (capacity), 20 ton yük miktarı (ton20), 30 ton yük miktarı (ton30) ve gemi yükü (shipLoad) gibi özellikleri bulunmaktadır. Ayrıca, ülkelerin adlarının karşılığında o ülkeye giden toplam, 20 ton ve 30 tonluk yük miktarlarını tutan bir sözlük olan countryDict’i oluşturur.

**def isFull(self):** Yük alanının kapasitesinin 20'den küçük olup olmadığını kontrol eder. Eğer kapasite 20'den küçükse, yük alanı dolmuş kabul edilir ve True döndürülür.

**def isEmpty(self):** Yük alanının kapasitesinin 750'ye eşit olup olmadığını kontrol eder. Eğer kapasite 750'ye eşitse, yük alanı boş kabul edilir ve True döndürülür.

**def load(self, truck):** Kamyon tarafından taşınan yükleri yük alanına yükler. Eğer yük alanı boşsa, istif alanının boş olduğuna dair bir mesajı yazdırır. Ek olarak tır istif alanına bir yükleme yaparsa, bunu kullanıcıya tırın özellikleriyle birlikte bildirir. Ayrıca, yük alanının kapasitesi tırın taşıdığı yük kadar azaltılır, 20 ton ve 30 ton yük miktarları ve ülke bilgileri güncellenir.

**def unload(self, ship**): Gemiden yük alanına yük boşaltır. Eğer gemi kapasitesi 30'dan büyükse ve 30 ton yük miktarı sıfırdan büyükse, 30 ton yük boşaltılır. Ek olarak eğer gemi istif alanından yük alırsa bunu kullanıcıya bildirir. Aynı şekilde, eğer gemi kapasitesi 20'den büyükse ve 20 ton yük miktarı sıfırdan büyükse, 20 ton yük boşaltılır. İlgili bilgiler güncellenir ve işlem sonucunda geminin doluluk durumu kontrol edilir.

**def printInfo(self):** Yük alanındaki ülkelerin bilgilerini ekrana renkli bir şekilde yazdırır. Toplam yük miktarı, 20 tonluk yük sayısı ve 30 tonluk yük sayısı gösterilir. Çıktı aşamasında bu fonksiyon kullanılmaz. Sadece kod yazarken programcıya ek bir görüş sağlamak amacıyla yazılmıştır.

## Group() fonksiyonu:

Kendisine parametre olarak gelen iç içe listeyi, iç listesinin indisi aynı olan elemanlara ayırıp; ayrı bir iç listeye atmaya yarar (bir nevi bunları gruplar).

Bunu yaparken önce bir sözlük yapısı oluşturur ve eğer ilk elemanlar aynıysa bu yapıya (aynı olan eleman: tüm liste) şeklinde bu yapıya ekler.

En son olarak bu sözlük yapısını farklı bir listeye yerleştirir. Böylece üç katmanlı ve ilk indise göre gruplanmış bir liste elde eder.

## Order() fonksiyonu:

Kendisine parametre olarak gelen iki string değeri ve bir default parametreyi (bu parametreler, dosya dizinleri ve gruplama yapılıp yapılmayacağını belirten parametrelerdir.), kullanarak ödevde istendiği gibi önce birinci sonra ikinci indise göre sıralar.

Bu işlemi yaparken önce open fonksiyonu yardımıyla (şifrelemesi windows-1254 olacak şekilde) string değerleri açılmış dosyaya çevirir.

Peşinden bu dosyaları for döngüleriyle okur ve satırları bir listeye atar. Liste içerisindeki elemanların istenmeyen karakterleri silindikten sonra

sorted() fonksiyonu yardımıyla değerleri önce birinci sonra ikinci indise göre sıralar. Sonrasında tırlar için ayrılan listeyi yukarıdaki group fonksiyonu yardımıyla

gruplara ayırır (eğer group= True seçildiyse) ve elde edilen listeleri geri döndürür.

## Simulation() fonksiyonu:

Bu bölümde, simulation fonksiyonu içinde gemilerin yüklenme ve boşaltılma işlemlerini gerçekleştiren bir döngü bulunmaktadır. Bu döngü, gemilerin varış zamanına göre işlem yapar ve gemilere ait yükleri uygun şekilde taşıma işlemlerini gerçekleştirir.

İlk olarak, gemi listesi (shipList) kontrol edilir. Eğer bu liste boş değilse, bir gemi var demektir. Bu durumda, gemiye ait bilgiler alınarak (shipAttrs), bir Ship nesnesi oluşturulur (ship). Ardından, bu gemi için bir iç döngü başlar.

İç döngüde, cargoBay1 ve cargoBay2'de bulunan yükleri gemiye yüklemek için işlemler gerçekleştirilir. Her iki istif alanı da gemiye yükleyebilecek durumdaysa, önce cargoBay1'den sonra da cargoBay2'den yüklemeler yapılır. Bu işlemler, istif alanlarındaki yükleri gemiye aktarmak için CargoBay sınıfının unload üye fonksiyonu kullanılarak gerçekleştirilir.

İç döngü, geminin kapasitesine ulaşıncaya veya istif alanlarındaki yükler tamamen bittikten sonra sona erer. Eğer gemi tamamen yüklendiyse, shipList’ten bu gemi çıkarılır ve ship değişkeni None olarak değiştirilip limandan uzaklaşır. Bundan sonra fonksiyon bir sonraki geminin işlemlerine geçer. Eğer istif alanları boşaldıysa veya geminin kapasitesi tamamlanmadıysa, iç döngüden çıkılır ve bir sonraki zaman adımına geçilir.

* 1. **Main() fonksiyonu:**

Bu fonksiyon programın başlığını ve kullanıcının programla etkileşime girmesi için kullanması gereken girdileri görmesini sağlayan bir menü yazdırır.

Bunu yaparken öncelikle sonsuz döngüye başlaması için “control” adında bir değişkeni True değerine atar. Bu değer kullanıcı çıkış girdisini girene kadar True olarak kalır ve kullanıcının programı devamlı olarak kullanmasını sağlar. Eğer kullanıcı “Simüle et” seçeneğini girerse; program, verilen dosyalar ile liman otomasyonunu yazdırır. Eğer kullanıcı “Tır bilgilerine eriş” seçeneğini girerse; program bu sefer, Ulaşma saati ve plakası verilen tırın bilgilerini (dosyadan okuyup bir sözlük yapısına atayarak) ekrana bastırır. Eğer kullanıcı “Gemi bilgilerine eriş” seçeneğini girerse; program, İsmi verilen geminin bilgilerini ekrana bastırır. Eğer kullanıcı “Çıkış” seçeneğini seçerse; program, control değişkenini True’ya çevirip en baştaki döngüyü sonlandırır.

# SONUÇ VE ÖĞRENİLEN DERSLER

Bu proje sayesinde Python programlama dilinin kullanımı, büyük çaplı projelerin taslaklarının nasıl oluşturulabileceği ve bu projelere dair resmi evrakların nasıl tamamlanabileceği hakkında deneyim ve bilgi sahibi olduk. Ayrıca, ekip çalışması ve kod standartları konularına da dikkat ettik.

# KAYNAKÇA

<https://talyian.github.io/ansicolors/>

<https://www.w3schools.com/python/python_file_handling.asp>

<https://stackoverflow.com/questions/8527503/constant-lambda-function-in-python>

**6. GITHUB BAĞLANTILARI**

Yavuz Selim Gürsoy: <https://github.com/Yavuz-Selim-Gursoy/Liman-Otomasyonu>

Talha Tuna: https://github.com/TalhaTuna2/Liman-Otomasyonu