

**T.C**

**KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**LİMAN OTOMASYONU**

**Hazırlayan**

**EMRAH ŞAHİN MÜGE ÖZLER**

220502025 210501004

**DERS SORUMLUSU**

**PRF.DR. HÜSEYİN TARIK DURU**

**13 ARALIK 2023**

**İÇİNDEKİLER**

[1.](#_1fob9te) ÖZET 1

[2.](#_3znysh7) GİRİŞ 2

[3.](#_2et92p0) YÖNTEM 3

[4.](#_2jxsxqh) SONUÇ VE ÖĞRENİLEN DERSLER 4

[5.](#_z337ya) KAYNAKÇA 5

6. GİTHUB ADRESLERİ 6

# ÖZET

Bu projedeki amacımız kodlama dillerinden biri olan Python’da gemiler ve olaylar dosyalarını kullanarak bir liman otomasyonunu oluşturmaktır. Bu otomasyonu oluştururken yığın yapısını kullanmamız önerilmiş olup yığın yapısı kullanılarak ödev yapılmıştır.

# GİRİŞ

Bu projede kullanılması ve takip edilmesi gereken iki dosyamız mevcuttur. Bu dosyalarda iki farklı durum gösterilmiştir. Gemiler dosyasında 4 ülkeye giden gemiler, taşıyabilecekleri yük miktarlarını ve limana geliş zamanlarını gösteren bilgiler yer almaktadır. Olaylar dosyasında ise limana gelen tırlar, bu tırların geliş zamanları, plakaları, taşıdıkları yük miktarları, bu yüklerin hangi ülkeye gidecekleri ve maliyetleri gibi durumların yer aldığı bilgiler yer almaktadır. Bu bilgilerden faydalanarak iki istif alanına sahip limanın vinç yardımıyla tırlardan alınan yükleri istif alanına daha sonra gemilere yüklenmesi ile gerçekleşebilecek bir liman işletim sisteminin oluşturulması istenmektedir.

# YÖNTEM

Her şeyden önce ilk olarak kullanacağımız modülleri dosyaya ekledik.

Bu aşamadan sonra ilk olarak sınıflarımızı oluşturduk. Tır sınıfının içinde nesnelerimiz tır plakası, tırın varış zamanı, yükün gideceği ülke, yirmi ve otuz tonluk konteyner adeti, taşıdığı yük miktarı, yükün maliyetini içeren nesne oluşturulur. Bir sözlük oluşturularak tır bilgileri sözlüğe aktarılır. Get-info fonksiyonu ile sözlük bilgileri döndürülür. Bu sayede bir nevi C++’da olduğu gibi veri kapsüllemesi yapılmak istenmiştir.

İkinci olarak gemi sınıfı oluşturulur. Gemi sınıfı içerisinde gemi adı, gideceği ülke, varış zamanı, kapasitesi, taşıdığı yük miktarı ilk olarak sıfır olacak şekilde tanımlanır ve bir sözlük oluşturulur. Bu kısımda tırlarla gelen yük gemiye taşındıkça geminin yükünü belirleyen ve her yükleme ile gemi yük miktarını güncelleyen bir fonksiyon tanımlanır. Yine aynı tır sınıfındaki gibi sözlüğe ulaşmamızı ve içindeki bilgileri getirmemizi sağlayan get-info fonksiyonu sınıf içerisinde tanımlanır.

Tır bilgilerini içeren dosyayı okumak için read fonksiyonun türevleri kullanılır ve bu fonksiyon içerisinde tırların plakaları, geliş zamanları, taşıdıkları yükler ve miktarları, hangi ülkeye gideceği ve maliyeti gibi ifadeler karakterler olarak chardet\_encording fonksiyonu çağırılarak dosya hangi karakter kodlamasını içeriyorsa (UTF-8 gibi) ona göre dosyayı okumamızı sağlar ve daha sonra diğer tanımladığımız, time\_and\_plaque\_sort\_of\_truck fonksiyonu ile tırlar geliş zamanlarındaki plakalarına göre sorted fonksiyonu yardımı ile sıralama yapılır. Bu sıralama daha sonra tırları okuma dosyası içinde önce listesi biçiminde tutulur sonra bu listede list comprehension yöntemi ile her elemanın bir nesnesi oluşturulur ve bu nesnelerin hepsi bir liste halinde tutularak fonksiyonun return değerini oluşturur. Böylece her satırı temsil eden nesne listesi oluşmuş olur. Bu işlem gemiler için biraz farklıdır. Sadece gemileri okunarak her satırı sözlük şeklinde olan bir liste haline gelmiştir. Ve bu listeyi yine read\_file\_truck fonksiyonunda olduğu gibi list comprehension yöntemi ile her liste elemanını bir nesnesini oluşturulur ve bu nesnelerin hepsi bir liste halinde tutularak fonksiyonun return değerini oluşturur. Böylece her satırı temsil eden nesne listesi oluşmuş olur.

Böyle fonksiyonlar dışında dosya globalinde kullanılmak için tanımlanan birkaç değişkenlerimiz var ve bu değişkenleri fonksiyon içinde global anahtar kelimesiyle çağırıp fonksiyonların içinde kullanılır.

Bu kısımdan sonra yük indirme ve yük yükleme kısımlarına geçiş yapılır. Burada yük indirme tırlar üzerinden olacağı için tırları nesne olarak alan fonksiyon yazılır. Bu fonksiyonu yük indirme şeklinde adlandırıldı. t zamanında gelen tırların ayrı listesini tutup istif alanı 1’e yığın şeklinde olacak şekilde indirilir yani ilk gelen tırın yükü en altta bulunur. Burada dikkat edilmesi gereken hususlar; istif alanlarının belli bir kapasitesi var, istif alanı 1’e sadece tırlardan gelen yükler iner, istif alanı 2 ise gemilere yükleme yaparken yığınları düzenlemek için kullanılacağıdır. Yani bir yığından yük almak istediğimizde eğer o yük yığının en üstünde değil de alttaysa üsteki yükleri istif alanı 2’ye koyar. Sonra o yükümüzü alır ve tekrar istif alanı 1’e koyar. Bu nedenle bize sürekli yükleme ve istif alanının kapasitesi hakkında bilgi geri dönütünün sağlanması sağlandı.

Yük yükleme işleminde kullanılan fonksiyon ise istif alanının en üstünden yığını alarak işe başlar ve yığın prensibine göre ilerler. Burada dikkat edilmesi gereken husus gemiler ile yüklerin doğru şekilde yüklenmesidir. Bu nedenle geminin yükü olmayan istif alanı 1 de bulunan yük, istif alanı 2 ye aktarılacaktır. Bu kısımda yine istif alanı 2 maksimum kapasitesine dikkat edilir. İstif alanı 2 dolunca veya maksimum kapasitesine ulaşınca uyarı olarak geri dönüt sağlanır.

Bu işlemlerin yanı sıra oluşan maliyet hesabının da yapılması gerekmektedir. Bu nedenle maliyet hesabını yapan bir fonksiyon yazıldı ve yük miktarlarıyla maliyetini hesaplayan kod yazıldı ve kod çalıştırmasının sonunda toplam maliyeti kullanıcıya gösterecek şekilde nesnesi oluşturulup yazılmıştır.

Simülasyon işlemini başlatan fonksiyon işlemi yazıldı. Bu fonksiyon, her bir zaman adımında gelen tırları kontrol eder, varsa yükleri indirir ve gelen tırlar listesine alıp ona göre nesne oluşturur. Ayrıca, o anki zamanda gelen gemileri kontrol eder ve bekleyen gemiler listesine ekler. Bu işlemler, belirli bir süre boyunca (greatest\_value + 1 zaman adımı) devam eder. Bu süre içinde gerçekleşen olayları ve durumları ekrana basar.

Bu simülasyon, liman otomasyon sistemini temsil eden bir modeldir ve belirli kurallar dahilinde gemi ve tır hareketlerini simüle eder. Bu kısım, istif alanı 1'deki uygun yükleri belirli bir ülkeye gidecek gemilere yüklemek için gereken işlemleri içerir. İstif alanı 1'deki yükleri kontrol eder, uygun gemiye yüklenirse ilgili geminin yük bilgisini günceller ve eğer gemi belirli bir doluluk oranına ulaşırsa gönderir. Eğer gemi doluysa veya yüklenemiyorsa, döngüyü kırar ve bir sonraki gemiye geçer. Bu işlemler sırasında geçici bir yığında yükleri taşıma işlemi de gerçekleştirilir. Daha sonra geçici yığındaki yükler istif alanı 1'e geri yüklenir. Son olarak, yüklenemeyen gemiler temizlenir ve sonuçlar ekrana yazdırılır.

# SONUÇ VE ÖĞRENİLEN DERSLER

Bu proje sayesinde Python dilinin kullanımı ve kullanırken dikkat edilmesi gereken hususları, istenilen yöntemlerin hesaplanma işlemi sırasında sınıf bloklarının oluşturulması nesnelerin neler olması gerektiği, parametreleri ve işlemlerin doğru olup olmadığının kontrolünün yapılması gibi işlem basamaklarını tanımlamak gerekliliği gibi kavramlar üzerinde duruldu. Bu kısımda kullanılan yığın yapısına uygun olarak yük yükleme ve gerekli yüklerin gemilere aktarılması gibi hususlara dikkat edilmesi gerektiğidir. En zorlanılan kısım bu kısım oldu çünkü nasıl yapılacağı konusunda diğer gruplar ve yazılan kodlar üzerinde araştırma yaparak uygulanmaya çalışıldı. Ödevimizde eksik olan yer ise vinç olayı yoktur. Yani ödevde istenen en fazla 20 vinç işlemi olayı yoktur. Kodumuz o an ne gerekiyorsa onu yapıyordur. İstif alanlarının doluluk oranları için kod satırları yazıldı ama bu vinç gereksinimini yapılamadığı için kodun çalışma zamanında herhangi bir istif alanı doldu bilgisi alamıyoruz. Yani kod gereken her şeyi o an yaptığı için istif alanları dolmadan boşaltılıyor ve her zaman yeni gelecek yük için yer oluyor. Bu eksik haricinde bir eksik fark edilememiştir.

# KAYNAKÇA

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

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

https://www.geeksforgeeks.org/python-dictionary/?ref=lbp

<https://www.w3schools.com/python/pandas/default.asp>

<https://realpython.com/list-comprehension-python/>

https://www.geeksforgeeks.org/python-classes-and-objects/

<https://stackoverflow.com/questions/13517080/sum-a-csv-column-in-python>

https://copyprogramming.com/howto/sum-a-csv-column-in-python

<https://www.geeksforgeeks.org/how-to-sort-data-by-column-in-a-csv-file-in-python/>

<https://www.tutorialspoint.com/how-to-sort-csv-by-a-single-column-in-python>

<https://realpython.com/python-encodings-guide/>

<https://stackoverflow.com/questions/37177069/how-to-check-encoding-of-a-csv-file>

# GitHub Adresleri

<https://github.com/emrahsahn/EmrahSahin>

<https://github.com/mugeozler/liman_otomasyonu>