T.C

KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

PROGRAMLAMA SUNUMU 2

Hazırlayan

UTKU EMRE ERMİŞ — **220501034**

GÖKHAN TURHAN — **220501002**

https://github.com/UtkuEmreErmis
https://github.com/GokhanTurhan55

DERS SORUMLUSU

NUR BANU ALBAYRAK

YAPTIKLARIMIZ

1. **TIR ve GEMİ SINIFLARI:**

“TIR” ve “Gemi” adlı iki sınıf tanımlanmıştır. Her biri kendi özellikleri ve yük bilgilerini içeren bir liste ile birlikte gelir.

a. “TIR” sınıfı:

“ __init__” metodu: TIR sınıfının yapıcı metodu. Bir TIR nesnesi oluşturulduğunda çağrılır.

plaka: TIR'ın plakasını temsil eden bir özelliktir.

yukler: TIR'ın taşıdığı yüklerin bilgilerini içeren bir liste.

b. “Gemi” sınıfı:

“__init__” metodu: Gemi sınıfının yapıcı metodu. Bir Gemi nesnesi oluşturulduğunda çağrılır.

gemi_numarasi: Geminin numarasını temsil eden bir özelliktir.

kapasite: Geminin taşıma kapasitesini temsil eden bir özelliktir.

ulke: Geminin gideceği ülkeyi temsil eden bir özelliktir.

yukler: Gemide taşınan yüklerin bilgilerini içeren bir liste.

2. “liman_simulasyonu” FONKSİYONU

“liman_simulasyonu” fonksiyonu, limandaki yük indirme-yükleme sürecini simüle eden ana fonksiyondur. Bu fonksiyon, belirli kurallar ve sınırlamalar altında TIR'lar ve gemiler arasındaki etkileşimi yönetir.

gemiler ve tirlar: Dosyadan okunan gemi ve TIR verilerini içeren sözlük yapısındaki bilgiler.

istif_alani: Limandaki toplam istif alanının ton cinsinden kapasitesini temsil eder (başlangıçta 750 ton).

vinc_kapasitesi: Bir seferde taşınabilecek maksimum yük miktarını belirten sabit (20 ton).

liman_takvimi: Limandaki olayların zaman çizelgesini saklamak için kullanılan bir sözlük. Her zaman anında gerçekleşen olayları içerir.

t: Simülasyonun çalıştığı anı temsil eder. Belirli bir zaman aralığında, hem gemilerin hem de TIR'ların olaylarını işler.

a. If t in tirlar

Belirli bir zaman anında “TIR”lar yük indirir. İlgili “TIR”ların plakalarına göre sıralanır ve plaka sırasına göre yük indirilir. İstif alanı doluysa beklenir.

b. If t in gemiler

Belirli bir zaman anında gemilere yük yüklenir. İlgili gemiler gemi numarasına göre sıralanır ve gemi numarasına göre yüklenir. İstif alanı boşsa beklenir.

3. “dosyadan_veri_oku” FONKSİYONU

“dosyadan_veri_oku” fonksiyonu, CSV dosyasından veri okuyarak, gemi ve TIR nesnelerini oluşturan ve bu nesneleri bir sözlük içinde zaman bazlı olarak düzenleyen bir işlemdir.

dosya_adi: Okunacak CSV dosyasının adını temsil eder.

veri: Okunan verilerin saklandığı bir sözlük. Bu sözlük, her bir zaman anındaki gemi ve TIR nesnelerini içerir.

with open(...) as dosya: Belirtilen dosyayı açar ve işlem bittiğinde otomatik olarak kapatır.

csv.DictReader(dosya): CSV dosyasını sözlük formatında okumak için bir DictReader nesnesi oluşturur.

for row in reader: Her bir satırı okur ve bir sözlük olarak row değişkenine atar.

zaman = int(row['geliş_zamanı']): Zaman bilgisini okur ve tamsayıya çevirir.

'tır_plakası' in row: Eğer 'tır_plakası' bilgisi row sözlüğünde varsa, bu bir TIR'nın bilgileridir.

plaka = row['tır_plakası']: TIR'ın plaka numarasını okur.

ulke = row['ülke']: TIR'ın yük taşıdığı ülke bilgisini okur.

ton20_adet = int(row['20_ton_adet']): 20 tonluk konteyner adedini okur ve tamsayıya çevirir.

ton30_adet = int(row['30_ton_adet']): 30 tonluk konteyner adedini okur ve tamsayıya çevirir.

yuk_miktar = int(row['yük_miktarı']): Yük miktarını okur ve tamsayıya çevirir.

maliyet = int(row['maliyet']): Yükün maliyetini okur ve tamsayıya çevirir.

tir = TIR(plaka): Oluşturulan bilgilerle yeni bir TIR nesnesi oluşturulur.

tir.yukler.append(...): TIR'ın yükler listesine, 20 tonluk ve 30 tonluk konteyner bilgilerini ekler.

veri[zaman].append(tir): Oluşturulan TIR nesnesini, ilgili zaman anındaki listeye ekler.

'gemi_adı' in row: Eğer 'gemi_adı' bilgisi row sözlüğünde varsa, bu bir gemi bilgisidir.

gemi_numarasi = int(row['gemi_adı']): Geminin numarasını okur ve tamsayıya çevirir.

kapasite = int(row['kapasite']): Geminin kapasitesini okur ve tamsayıya çevirir.

ulke = row['gidecek_ülke']: Geminin gideceği ülke bilgisini okur.

gemi = Gemi(gemi_numarasi, kapasite, ulke): Oluşturulan bilgilerle yeni bir Gemi nesnesi oluşturulur.

veri[zaman].append(gemi): Oluşturulan Gemi nesnesini, ilgili zaman anındaki listeye ekler.

return veri: Okunan verilerin düzenlenmiş hali olan sözlüğü döndürür.

4. UNICODE HATASINI GİDERME

“dosyadan_veri_oku” fonksiyonunda dosyayı açarken encoding="utf-8" parametresi eklenerek UnicodeDecodeError hatası giderilmiştir.

5. SENARYO UYGULAMASI

“__main__” bloğu içinde, "gemiler.csv" ve "olaylar.csv" dosyalarından veriler okunarak liman_simulasyonu fonksiyonu çağrılmıştır.

6. KÜTÜPHANELER

csv: Bu kütüphane, CSV dosyalarını okumak ve yazmak için kullanılır. “csv.DictReader” sınıfı, bir CSV dosyasını satır satır okumak için kullanılmıştır.

collections.defaultdict: Bu kütüphane, ekstra bir argüman olarak belirtilen varsayılan bir değere sahip bir sözlük sağlar. Programda, “defaultdict(list)” kullanılarak zaman anahtarlarına karşılık gelen değerlerin listelerini depolamak için kullanılmıştır.