Gazi Üniversitesi Nesne Yönelimli Programlama Dersi Proje Raporu

Proje İsmi: Lojistik Yönetim Sistemi

Ad-Soyad : Burak Fatih Özen

Öğrenci Numarası: 23181616012

Ad-Soyad: Mehmet Gökmenoğlu

Öğrenci Numarası: 23181616046

Ders Adı: Nesne Yönelimli Programlama (BMT-209)

Sınıf Bilgisi: 2.Sınıf

Dönem Bilgisi: Güz Dönemi

İÇİNDEKİLER

Giriş Bölümü

Projenin Amacı:

Kullanıcıyı ürününü kargoya teslim eden bir müşteri varsayarak kargolama sürecinde gereken bilgileri kullanıcı girdisi ile alıp gerekli metot ve sınıflar aracılığıyla işleyerek müşterinin ihtiyacı olan verileri sunmak.

Problemin Kapsamı:

Bu proje, lojistik ve taşımacılık sektöründe karşılaşılan maliyet ve zaman hesaplama zorluklarına çözüm sunmayı amaçlayan bir uygulamanın geliştirilmesini kapsamaktadır. Sistem, farklı taşıma yöntemlerini (hava ve kara taşımacılığı) destekleyecek, ürünlerin fiziksel özelliklerine ve öncelik durumlarına göre en uygun taşıma yöntemini önererek kullanıcılara karar desteği sağlayacaktır.

Projede Kullanılan Teknolojiler:

Java programlama dili kullanılarak nesne yönelimli programlama (OOP) prensiplerine dayalı bir yapı oluşturulmuş, kullanıcı girdisi ile veri işleme, sınıflar arası etkileşim ve hata yönetimi gibi temel işlevle rgerçekleştirilmiştir. Verilerin yönetimi ve saklanması için dosya okuma ve yazma işlemleri kullanılmış, kullanıcı dostu bir deneyim sunulmuştur. Uygulama, kullanıcıya kargo süreciyle ilgili bilgi sunmak amacıyla sınıflar ve metotlar aracılığıyla veri işleme mantığına dayalı olarak geliştirilmiştir. Ayrıca, hata yönetimi amacıyla exception handling teknikleri kullanılarak özel hata sınıfları ve try-catch blokları ile güvenli bir veri akışı sağlanmıştır. Mesafe hesaplamaları için Google Maps API'leri kullanılmıştır. Geocoding API aracılığıyla şehirlerin koordinatları alınmış, hava yolu mesafesi hesaplanırken Distance Matrix API kullanılarak kara yolu mesafesi elde edilmiştir. HTTP istekleri için OkHttp kütüphanesi kullanılarak API'lerle etkileşim sağlanmış ve Gson kütüphanesi ile JSON verileri işlenmiştir. Uygulama, kullanıcıdan alınan şehir bilgilerini kullanarak mesafeleri hesaplamakta ve uygun ulaşım yönteminin (kara yolu veya hava yolu) seçilmesine olanak tanımaktadır.

Teknik Bölüm

Problemin Tanımı:

Günümüz lojistik sektöründe, ürünlerin taşınması sırasında maliyet ve zaman hesaplamalarının doğru bir şekilde yapılması, müşteri memnuniyeti ve operasyonel verimlilik açısından büyük önem taşımaktadır. Ancak, mevcut sistemlerde taşıma yönteminin belirlenmesi, ürünlerin önceliklendirilmesi, maliyet ve zaman hesaplamalarının kullanıcı dostu bir şekilde yapılması zor ve zaman alıcı olabilmektedi.

Ayrıca, ürünlerin boyutları, ağırlıkları, hassasiyet durumları gibi parametrelerin dikkate alınarak doğru bir taşıma yöntemi ve maliyet analizi yapılmasını sağlayacak

Entegre bir sistem eksikliği bulunmaktadır. Bu durum,lojistik firmalarının kaynaklarını verimli kullanamamasına ve kullanıcıların taşıma seçenekleri arasında ideal bir seçim yapamamasına neden olmaktadır.

Çözümde Kullanılan Sınıflar, Metotlar ve Mimariler:

Çözümde 13 adet birbirinden farklı özelliklerde sınıf kullanıldı.

Kullanılan Sınıflar

(CustomExceptionClass)

Amaç: Özel bir hata mesajı oluşturmak için kullanılmıştır. Kullanıcı yanlış veri girdiğinde anlamlı bir uyarı mesajı göstermek için tasarlanmıştır.

Metotlar:

Varsayılan hata mesajı içeren bir constructor.Kullanıcı tarafından sağlanan özel bir hata mesajı içeren constructor.

(AirTransportationCost)

Amaç: Hava taşımacılığına ait maliyet, kapasite, taşıma süresi gibi hesaplamaları yapmak.

Özellikler:

Hava taşıma yöntemi, hız, kapasite ve maliyet hesaplamaları.

Metotlar:

calculateCostPerKm(): Hava taşımacılığının km başına maliyetini hesaplar.

calculateCapacity(): Hava taşıma kapasitesini döner.

totalCostCalculator(): Toplam maliyeti hesaplar.

cargoFee(): Kargo ücretini hesaplar.

TimeCalculator(): Hava taşıma süresini hesaplar.

(RoadTransportationCost)

Amaç: Karayolu taşımacılığına ait maliyet, kapasite ve taşıma süresi hesaplamalarını yapmak.

Özellikler:

Karayolu taşıma yöntemi, hız, kapasite ve maliyet hesaplamaları.

Metotlar:

calculateCostPerKm(): Karayolu taşımacılığının km başına maliyetini hesaplar.

calculateCapacity(): Karayolu taşıma kapasitesini döner.

totalCostCalculator(): Toplam maliyeti hesaplar.

cargoFee(): Kargo ücretini hesaplar.

TimeCalculator(): Karayolu taşıma süresini hesaplar.

(Packaging)

Amaç: Ürün paketlemesi için maliyet hesaplamalarını yapmak.

Özellikler:

Ürün boyutu, ağırlığı ve hassasiyet bilgileri.

Metotlar:

productSizeMultipiler(): Boyut bazında katsayı döner.

productWeightMultipiler(): Ağırlık bazında katsayı döner.

packageCostCalculator(): Paketleme maliyetini hesaplar.

(Product)

Amaç: Ürünün kategorisini ve önceliğini tanımlamak.

Özellikler:

Ürün kategorisi ve öncelik düzeyi.

Metotlar:

getCategory(): Ürün kategorisini döner.

getPriority(): Ürün öncelik düzeyini döner.

(Storage)

Amaç: Depolama alanını ve ürünlerin stok durumunu yönetmek.

Özellikler:

Ürün kategorisi, stok durumu ve öncelik bilgileri.

Metotlar:

productCategoryControl(): Ürün kategorisine göre öncelik belirler.

(DistanceCalculator)

Amaç: Ulaşım mesafelerini hesaplamak için statik metotlar sağlar.

Metotlar:

getAirDistance(): Hava mesafesini döner.

getRoadDistance(): Karayolu mesafesini döner

(TimeCalculator (Abstract Class))

Amaç: Taşıma sürelerinin hesaplanmasını standartlaştırmak.

Metotlar:

TimeCalculator(): Hızın km/saat veya m/s cinsinden süreyi hesapladığı soyut metotlar.

(Transportable (Interface))

Amaç: Ulaşım türleri için maliyet ve kapasite hesaplamalarını standartlaştırmak.

Metotlar:

calculateCostPerKm(): Araç başına km maliyetini döner.

calculateCapacity(): Araç kapasitesini döner.

(TransportMethod (Enum))

Amaç: Kullanılabilir ulaşım yöntemlerini tanımlamak.

Değerler:

ROAD: Karayolu taşımacılığı.

AIR: Hava taşımacılığı.

(InputClass)

Özellikler (Attributes)

categories (ArrayList<String>): Kullanıcının girdiği ürün kategorilerini saklar.

priorities (ArrayList<Integer>): Ürünlerin öncelik düzeylerini saklar

Metotlar

inputMethod()

Amaç: Kullanıcıdan ürün bilgilerini almak ve gerekli işlemleri gerçekleştirmek.

İşlevleri:

Kullanıcıdan ad alır (Çıkış için 'quit' seçeneği).

Kullanıcıdan ürün kategorisi, boyutu, ağırlığı ve hassasiyeti gibi bilgileri toplar.

Kategorilere ve önceliklere dayalı sıralama yapar.

Mesafe hesaplayıcı sınıfını (DistanceCalculator) kullanarak taşıma yöntemini belirler.

Tüm bilgileri dosyaya kaydeder ve sonuçları ekrana yazdırır.

sortAndDisplayProducts()

Amaç: Ürün kategorilerini ve önceliklerini sıralar ve ekrana yazdırır.

İşleyiş:

Product sınıfı kullanılarak kategoriler ve öncelikler bir listeye eklenir.

Listenin önceliğe göre sıralanması yapılır.

Sıralı liste ekrana yazdırılır.

fileMethod(String name, ArrayList<String> categories, ArrayList<Integer> priorities, String transportMethod)

Amaç: Kullanıcıdan alınan ve hesaplanan bilgileri bir dosyaya yazmak.

İşleyiş:

Kullanıcı bilgilerini ve ürün kategorilerini dosyaya kaydeder.

Hava ve karayolu taşımacılığı maliyetlerini hesaplayarak dosyaya yazar.

Kullanıcıdan hız birimini alarak taşıma süresini hesaplar ve dosyaya ekler.

readFile()

Amaç: Daha önce dosyaya kaydedilmiş bilgileri okumak ve ekrana yazdırmak.

İşleyiş:

BufferedReader kullanılarak dosyadan satır satır okuma işlemi yapılır.

Okunan her satır ekrana yazdırılır.

Mimariler

Statik Değişken ve Metot Kullanımı:

DistanceCalculator sınıfındaki statik metotlar kullanılarak taşıma yöntemi belirleniyor.

Modülerlik:

Farklı işlevler için ayrı metotlar tanımlanmış. Bu, sınıfın tek sorumluluk prensibine uygun olmasını sağlar.

Hata Yönetimi:

Kullanıcının yanlış veri girişi yapmasını önlemek için try-catch blokları ve özel hata sınıfı (CustomException) kullanılmış.

Dosya İşlemleri:

BufferedWriter ve BufferedReader kullanılarak bilgilerin dosyaya yazılması ve okunması sağlanmış.

Veri Yapıları:

ArrayList kullanılarak kategoriler ve öncelikler dinamik olarak saklanıyor.

(Storage)

Ürün kategorisinin önceliğini hesaplar.

productCategoryControl() metodu çağrılarak ürün önceliği belirlenir.

(Packaging)

Ürünün boyut, ağırlık ve hassasiyet bilgilerini işler.

setProductSize, setProductWeight, setProductSensitivity metotlarıyla ürün bilgileri ayarlanır.

(CustomException)

Kullanıcı hatalarını yönetmek için özelleştirilmiş istisna sınıfıdır.

(Product)

Ürünlerin kategori ve öncelik bilgilerini saklamak için kullanılır.

(TransportMethod)

Enum Sınıfı:

TransportMethod bir **Enum** sınıfıdır. Enum sınıfı, sabit değerleri grup haline getirip tanımlamak için kullanılır.

Enum'lar, bir türdeki sabit sayıda değeri temsil etmek için idealdir. Bu durumda, taşıma yöntemleri için iki sabit değer tanımlanmıştır: **ROAD** (Karayolu) ve **AIR** (Hava yolu).

Sabit Değerler:

ROAD: Karayolu taşımacılığını ifade eder.

AIR: Hava yolu taşımacılığını ifade eder.

Sonuç ve Değerlendirme

```
* Welcome to the Logistic Management System :)     *
* This program will help you about follow  process *
* of your cargo, calculating of your payment.      *
* You can use it by more than one person. It's     *
* sensible at multiple usage.                      *

Welcome! Please enter your name (or type 'quit' to exit):
Ayberk
Enter the category of your product
(Medicine-Foods-Valuable-Electronics-Industrial-Furniture-Textile-Other):
textile
You entered a valid category: textile
Enter the size of your product (cm3):
10000
Enter the weight of your product (kilograms):
12
Enter the sensitivity of your product ('yes' or 'no'):
no
Category: textile, Priority: 7
Do you want to add another product? 'yes' or 'no':
yes
Enter the category of your product
(Medicine-Foods-Valuable-Electronics-Industrial-Furniture-Textile-Other):
electronics
You entered a valid category: electronics
Enter the size of your product (cm3):
570
Enter the weight of your product (kilograms):
1.2
```

```
Enter the weight of your product (kilograms):
1.2
Enter the sensitivity of your product ('yes' or 'no'):
yes
Category: electronics, Priority: 4
Do you want to add another product? 'yes' or 'no':
no

Sorted list of products by priority:
Category: electronics, Priority: 4
Category: textile, Priority: 7
Boarding City: Ankara
Destination City: İzmir
Road Distance: 588.35 km
Air Distance: 520.47 km
Press '1' for cheapest way
Press '2' for fastest way
1
You have selected Road Transportation.
Enter the speed unit:km/h
Finally block is always executed.
Current Time is before the Updated Time.
Datas has registered to file.logistic.txt
```

```
Registered Datas in file:

Person name: Ayberk
Category: electronics, Priority: 4

Category: textile, Priority: 7

Transportation Method: ROAD
Total transportation cost: 882.52$
Cargo fee: 6.49$

Flight Speed (km/h): 90.0km/h
Flight Time: 6 Hour 32 Minute 13 Second
Current Time and Date: 23/12/2024 11:31:05
Arriving Time and Date: 23/12/2024 18:03:18

Welcome! Please enter your name (or type 'quit' to exit):
```

Sonuç olarak lojistik yönetiminde kullanıcı odaklı bir sistem oluşturuldu ve bununla beraber bir kullanıcının isteyebileceği tüm verilerin hesabı yapılarak kullanıcıya sunuldu

Öneri olarak ise Java swing kütüphanesi kullanılarak bu program bir arayüze aktarılabilir. Daha kullanıcı odaklı ve okunabilir olacaktır

Zorluklar ve Çözüm Yolları

API ile ilgili zorluklar:

Kullanıcının girdiği iki şehir arasındaki mesafeyi hava yolu ve kara yolu olarak hesaplamak amacıyla Google Distance Matrix API ve Google Geocoding API kullanmak istedik. Fakat API'ları programımıza entegre etme kısmında birçok sorunla karşılaştık. Örneğin kara yolu ve hava yolunu aynı yazdırıyordu veya bazı şehirleri görmüyordu program.

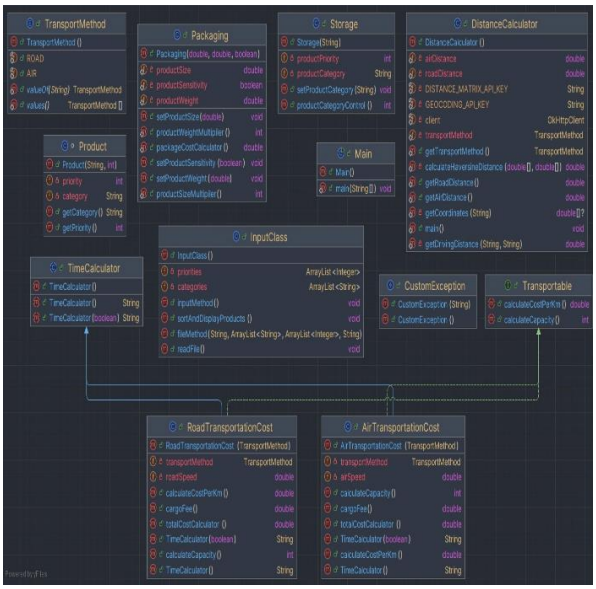
Aynı zamanda en son çalışır hale getirdikten sonra farklı bilgisayarda denediğimizde yine birçok sorun ortaya çıkıyordu.

Maven ile ilgili zorluklar:

API ile gerçek verileri projemize entegre etmek için Maven otomasyon aracını kullanmak zorundaydık fakat bu o kadar kolay olmadı. Kodların hepsi düzgün olmasına rağmen kod sayfasında maven ile ilgili 30'dan fazla hata ortaya çıkıyordu ve program çalışmıyordu.

Bununla ilgili uzun bir araştırma yaptıktan sonra bilgisayarda Maven'in kurulumuyla alakalı sorunlar olduğunu fark ettik. Tamamen Maven ve JDK'yı sıfırdan kurduktan ve bunların bin uzantılı dosyalarını PATH'e ekledikten sonra bu sorunların da üstesinden geldik

Ekler:



Kaynakça:

[1] İ. A. Doğru, "Dersler," Ders Sunumları. [Online]. Available: http://www.alperdogru.com.tr/dersler/. [Accessed: Nov. 22, 2024].

[2] GeeksforGeeks, "GeeksforGeeks," [Online]. Available: https://www.geeksforgeeks.org/. [Accessed: Dec. 22, 2024].

[3] OpenAI, "ChatGPT," [Online]. Available: https://chat.openai.com/. [Accessed: Dec. 22, 2024].

[4] Kodlama Vakti, "Kodlama Vakti Kanalı," YouTube, [Online]. Available: https://www.youtube.com/@KodlamaVakti/videos. [Accessed: Dec. 22, 2024].