T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

JAVA İLE NESNE YÖNELİMLİ UZAY SİMÜLASYONU GELİŞTİRME

B231210101 - Eyüp Mutlu Erol

SAKARYA

Nisan, 2025

Programlama Dillerinin Prensipleri Dersi

JAVA İLE NESNE YÖNELİMLİ UZAY SİMÜLASYONU GELİŞTİRME

Eyüp Mutlu Erol

B231210101 - 1.Öğretim C Grubu

Özet

Bu ödev kapsamında, farklı gezegenler arasında uzay araçlarının seyahatini simüle eden bir Java programı geliştirilmiştir. Temel problem, farklı zaman akış hızlarına (günlük saat sayıları) ve başlangıç tarihlerine sahip gezegenleri, bu gezegenler arasında belirli kalkış/varış tarihlerine ve yolculuk sürelerine sahip uzay araçlarını ve bu araçlarda ömürleri saatlik olarak azalan yolcuları yönetmektir. Veriler (gezegen, araç, kişi bilgileri) TXT dosyalarından okunmaktadır. Çözüm için Nesne Yönelimli Programlama (OOP) yaklaşımı benimsenmiş; Gezegen, Zaman, UzayAraci, Kisi gibi temel varlıklar için sınıflar tasarlanmıştır. Zaman yönetimi için Java'nın java.time.LocalDate sınıfı kullanılarak gerçekçi takvim ilerlemesi (artık yıllar, farklı ay uzunlukları dahil) sağlanmış ve zaman ilerlemesi ayrı bir Zaman sınıfında kapsüllenmiştir. DosyaOkuma sınıfı veri okuma ve nesne oluşturma işlemlerini, Simulasyon sınıfı ise ana simülasyon döngüsünü, durum güncellemelerini (kalkış, varış, ömür azalması, imha) ve konsol çıktısını yönetmektedir. Başlangıçta her araç için varış tarihi, gezegenlerin farklı zamanlarını dikkate alan bir "mini simülasyon" ile hesaplanmıştır. Konsol çıktısı, her adımda ekran temizlenerek güncel durumu formatlı bir şekilde göstermektedir

Anahtar Kelimeler: Java, Nesne Yönelimli Programlama, Simülasyon, Uzay Seyahati, Zaman Yönetimi, Dosya Okuma

1. GELİŞTİRİLEN YAZILIM

1.1 İstenilenler ve Projenin Amacı

Bu ödev kapsamında bizden istenilen, belirli kurallar çerçevesinde gezegenler arası uzay yolculuklarını simüle eden bir Java programı geliştirmekti. Temel gereksinimler şunlardı: Gezegen, uzay aracı ve kişi bilgilerini TXT dosyalarından okumak; her gezegenin kendine özgü gün uzunluğunu ve başlangıç tarihini dikkate alarak zamanı ilerletmek; araçların belirli tarihlerde kalkış yapmasını sağlamak; yolcuların ömürlerini saatlik olarak azaltmak ve ömrü bitenleri veya tüm mürettebatı ölen gemileri simülasyondan çıkarmak (imha etmek); araçlar yoldayken veya bir gezegende beklerken/varmışken nüfus sayımlarını doğru yapmak; simülasyonun anlık durumunu her saatlik adımda ekranı temizleyerek konsola yazdırmak ve tüm araçlar hedefine vardığında veya imha olduğunda simülasyonu sonlandırmak.Bu gereksinimler doğrultusunda projenin amacı, OOP prensiplerini kullanarak modüler, anlaşılır ve belirtilen kurallara uygun çalışan bir simülasyon yazılımı ortaya koymaktı. Aynı zamanda Java'nın standart kütüphanelerinden (özellikle dosya işlemleri için java.io ve tarih/zaman için java.time) etkin bir şekilde faydalanmak hedeflendi.

1.2 Ödevde Yapılanlar: Tasarım ve Sınıflar

Ödevin gerçekleştirilmesinde OOP yaklaşımı temel alındı ve aşağıdaki sınıf yapısı oluşturuldu:

Kisi: Yolcuları temsil eder; isim, yaş, kalan ömür (saat), ait olduğu araç adı ve hayatta olup olmadığını tutar. saatIlerle metodu ömrü azaltır.

Zaman: Bir gezegenin zamanını (LocalDate ve saat) ve gün uzunluğunu tutar, saatIlerle metodu ile zamanı doğru ilerletir. Mini simülasyon için kopyalanabilir yapıdadır.

Gezegen: Gezegen bilgilerini (ad, gün uzunluğu, nüfus) ve bir Zaman nesnesi referansını tutar. Zaman ilerletme gibi işlemleri Zaman nesnesine devreder. Mini simülasyon için başlangıç zamanının kopyasını sağlar.

AracDurumu: Araç durumlarını (BEKLIYOR, YOLDA, VARDI, IMHA) temsil eden bir enum'dır.

UzayAraci: Gemi bilgilerini, rotasını, durumunu, kalan yolculuk süresini ve başlangıçta hesaplanan varış tarihini tutar. Kalkış (kalkisYap), yolda ilerleme (saatIlerleYolda), imha olma (imhaEt) gibi durum değişikliklerini ve ilgili bilgileri yönetir. Nüfus sayımının doğru yapılması için nufusEklendiMi bayrağını içerir.

DosyaOkuma: TXT dosyalarını okur, verileri ayrıştırır, formatları (d.M.yyyy dahil) kontrol eder, hata yönetimi yapar ve nesneleri oluşturur.

Simulasyon: Ana simülasyon motorudur. Verileri alır, başlangıç ayarlarını yapar (nüfus, önceden varış tarihi hesaplama), ana döngüyü çalıştırır. Döngü içinde zamanı ilerletir, ömürleri azaltır, nüfusu günceller (varış sonrası ölümleri de dikkate alarak), araç durumlarını yönetir (kalkış, varış, imha) ve yazdirDurum ile formatlı çıktıyı üretir.

Main: Programın başlangıç noktasıdır; veri okumayı tetikler ve simülasyonu başlatır, sonunda toplam süreyi yazar.

1.3 Karşılaşılan Zorluklar ve Uygulanan Çözümler

Projenin geliştirilmesi sırasında bazı zorluklarla karşılaşıldı ve bunlar için çözümler üretildi:

- Gerçekçi Olmayan Takvim: Başlangıçta tarihleri basitçe ilerletme fikri, farklı ay uzunlukları ve artık yıllar nedeniyle sorunluydu. Bu zorluk, Java'nın java.time.LocalDate sınıfı kullanılarak aşıldı. Bu sınıf, takvim matematiğini otomatik olarak doğru bir şekilde yapar. Zaman yönetimi Zaman sınıfında kapsüllendi.
- Doğru Varış Tarihini Başta Hesaplama: Farklı zaman dilimlerindeki gezegenler arasında seyahat eden bir aracın varış tarihini en başta doğru hesaplamak zorluydu. Basit tahminler, simülasyonun gerçek sonucuyla tutarlı olmuyordu. Çözüm olarak, her araç için başlangıçta bir "mini simülasyon" yapan (hesaplaTumVarisTarihleriOnceden) bir yöntem geliştirildi. Bu yöntem, kalkışa kadar geçen süreyi ve bu sürede varış gezegeninin ilerlemesini hesaba katarak daha doğru bir başlangıç hesabı üretti. Bu yaklaşım kod karmaşıklığını ve başlangıç süresini artırsa da, doğruluk hedefine ulaşmak için tercih edildi.
- Nüfus Sayımı Tutarlılığı: Araçlar vardıktan sonra içindeki kişilerin ölmeye devam etmesi, nüfus sayımının tutarsızlaşmasına neden oluyordu. Bu sorun, bir kişi öldüğünde bulunduğu geminin durumu kontrol edilerek ve eğer gemi "Vardı" ise ilgili gezegenin nüfusunun azaltılmasıyla çözüldü. Ayrıca, varışta nüfusun sadece bir kez artırılması için UzayAraci sınıfına ek bir kontrol bayrağı (nufusEklendiMi) eklendi.
- İmha Mantığı: Bir geminin ne zaman imha olacağı (yolda mı, vardıktan sonra mı, başlangıçta mı) konusundaki farklı senaryoları doğru yönetmek için imha kontrol mantığı birkaç kez gözden geçirildi ve son olarak "yaşayan yolcusu kalmayan ve henüz imha olmamış her gemi imha olur" kuralı benimsendi.
- Konsol Yönetimi: Simülasyon çıktısının sürekli akması ve okunaksız olması, ekranTemizle fonksiyonu ve çıktıyı formatlayan (yazdirDurum) metotlarla aşıldı. Çok sayıda gezegenin ekrana sığdırılması için gezegen bilgileri gruplar halinde yazdırıldı.

1.4 Öğrenilenler

Bu ödev sürecinde aşağıdaki konularda önemli bilgiler ve deneyimler edinilmiştir:

- OOP Tasarım: Problemi soyutlayarak ilgili sınıfları (Varlıklar: Kişi, Gezegen, UzayAracı; Yönetim: Zaman, DosyaOkuma, Simulasyon) tasarlama ve bu sınıflar arasındaki ilişkileri kurma pratiği yapıldı. Kapsüllemenin önemi anlaşıldı.
- Java java.time API: LocalDate ve DateTimeFormatter sınıflarının tarih ve zaman işlemlerini ne kadar kolaylaştırdığı ve takvim karmaşıklıklarını nasıl soyutladığı öğrenildi.
- Simülasyon Mantığı: Zamanı adım adım ilerletme, durumları (state) yönetme, olayları (event) tetikleme (kalkış, varış, ölüm, imha) gibi temel simülasyon prensipleri uygulandı. Eş zamanlı ilerleyen farklı zaman akışlarını yönetmenin zorlukları görüldü.
- Algoritma Geliştirme: Özellikle doğru varış tarihini önceden hesaplama algoritması ("mini simülasyon") gibi problemlere çözüm üretme becerisi geliştirildi.
- Dosya İşlemleri ve Hata Yönetimi: TXT dosyalarından veri okuma, veriyi ayrıştırma (split), tip dönüşümleri (Integer.parseInt, LocalDate.parse) ve olası hataları (IOException, NumberFormatException, DateTimeParseException) try-catch ile yönetme pratiği yapıldı.
- Problem Çözme ve Hata Ayıklama: Beklenmedik çıktılarla karşılaşıldığında (örn: TestGemi'nin erken imhası gibi görünen durum), sorunun kaynağını bulmak için mantık yürütme, veriyi kontrol etme ve adım adım takip etme (gerektiğinde debug mesajları ile) yöntemleri kullanıldı.

2. ÇIKTILAR

İlk görselde programın başlangıç durumunun ekran görüntüsü , ikinci görselde ise sonlandıktan sonraki görüntüsü bulunmaktadır . UzunYolGemi adlı araç yola çıkmak için uzun bir süre beklemekte ve uzun saatler yol gitmektedir . VardiSonraImhaGemi aracı vardıktan sonra program bitmeden içindeki herkes öldüğü için vardıktan sonra imha olmaktadır . YoldaImhaGemi yoldayken içindeki herkes öldüğü için yolda imha olmaktadır. Diğer gemiler standart bir şekilde vardı sonucunu alan araçlardır . Program her türlü senaryoyu doğru çalıştıracak şekilde tasarlanmıştır.

```
--- BAŞLANGIÇ DURUMU ---
GEZEGENLER
          --- Dunya ---        --- Mars ---       --- Europa ---       --- Titan ---       --- Kepler186f ---
Tarih          01.01.2025        05.01.2025        10.01.2025        15.01.2025        01.02.2025
Nüfus               7                 6                 4                 7                 4

       --- Gliese581g ---    --- Pandora ---    --- Arrakis ---    --- Solaris ---    --- LV426 ---
Tarih          05.02.2025        10.02.2025        15.02.2025        20.02.2025        25.02.2025
Nüfus               2                 3                 2                 2                 3

Uzay Araçları:
Araç Adı             Durum          Çıkış          Varış          Hedefe Kalan Saat          Hedefe Varacağı Tarih
Gemi-A1              Bekliyor       Dunya          Mars           300                        17.01.2025
Gemi-A2              Bekliyor       Mars           Europa         400                        15.01.2025
Gemi-A3              Bekliyor       Europa         Titan          500                        16.01.2025
Gemi-A4              Bekliyor       Titan          Dunya          1000                       27.02.2025
Gemi-K1              Bekliyor       Kepler186f     Gliese581g     250                        15.02.2025
Gemi-K2              Bekliyor       Gliese581g     Pandora        200                        20.02.2025
Gemi-P1              Bekliyor       Pandora        Arrakis        350                        25.02.2025
Gemi-P2              Bekliyor       Arrakis        Solaris        450                        02.03.2025
Gemi-S1              Bekliyor       Solaris        LV426          600                        29.03.2025
Gemi-S2              Bekliyor       LV426          Dunya          1200                       20.02.2025
Gemi-X1              Bekliyor       Dunya          Europa         800                        20.01.2025
Gemi-X2              Bekliyor       Mars           Titan          650                        17.01.2025
Gemi-X3              Bekliyor       Europa         Pandora        550                        26.03.2025
Gemi-X4              Bekliyor       Titan          Arrakis        900                        04.05.2025
Gemi-X5              Bekliyor       Kepler186f     Solaris        700                        09.03.2025
UzunYolGemi          Bekliyor       Titan          Kepler186f     800                        07.04.2025
VardiSonraImhaGemi   Bekliyor       Mars           Europa         100                        11.01.2025
YoldaImhaGemi        Bekliyor       Dunya          LV426          500                        24.03.2025
```

```
--- SON DURUM (Saat: 2819) ---
GEZEGENLER
          --- Dunya ---        --- Mars ---       --- Europa ---       --- Titan ---       --- Kepler186f ---
Tarih          28.04.2025        27.04.2025        12.02.2025        22.01.2025        05.05.2025
Nüfus               6                 3                 4                 4                 2

       --- Gliese581g ---    --- Pandora ---    --- Arrakis ---    --- Solaris ---    --- LV426 ---
Tarih          16.05.2025        18.06.2025        04.05.2025        19.04.2025        15.07.2025
Nüfus               2                 4                 5                 4                 2

Uzay Araçları:
Araç Adı             Durum          Çıkış          Varış          Hedefe Kalan Saat          Hedefe Varacağı Tarih
Gemi-A1              Vardı          Dunya          Mars           0                          17.01.2025
Gemi-A2              Vardı          Mars           Europa         0                          15.01.2025
Gemi-A3              Vardı          Europa         Titan          0                          16.01.2025
Gemi-A4              Vardı          Titan          Dunya          0                          27.02.2025
Gemi-K1              Vardı          Kepler186f     Gliese581g     0                          15.02.2025
Gemi-K2              Vardı          Gliese581g     Pandora        0                          20.02.2025
Gemi-P1              Vardı          Pandora        Arrakis        0                          25.02.2025
Gemi-P2              Vardı          Arrakis        Solaris        0                          02.03.2025
Gemi-S1              Vardı          Solaris        LV426          0                          29.03.2025
Gemi-S2              Vardı          LV426          Dunya          0                          20.02.2025
Gemi-X1              Vardı          Dunya          Europa         0                          20.01.2025
Gemi-X2              Vardı          Mars           Titan          0                          17.01.2025
Gemi-X3              Vardı          Europa         Pandora        0                          26.03.2025
Gemi-X4              Vardı          Titan          Arrakis        0                          04.05.2025
Gemi-X5              Vardı          Kepler186f     Solaris        0                          09.03.2025
UzunYolGemi          Vardı          Titan          Kepler186f     0                          07.04.2025
VardiSonraImhaGemi   İMHA           Mars           Europa         --                         --
YoldaImhaGemi        İMHA           Dunya          LV426          --                         --
```

3. SONUÇ

Geliştirilen Java programı, metin dosyalarından alınan verilere dayanarak karmaşık uzay yolculuğu senaryolarını başarıyla simüle etmektedir. Nesne Yönelimli yaklaşım, projenin modülerliğini ve anlaşılabilirliğini sağlamıştır. java.time API'sinin etkin kullanımı ile farklı zaman akışlarına sahip gezegenlerin yönetimi gerçekçi bir şekilde ele alınmıştır. Özellikle başlangıçta doğru varış tarihlerini hesaplamak için geliştirilen "mini simülasyon" algoritması, projenin doğruluğunu artıran önemli bir adımdır. Nüfus takibi ve farklı imha senaryolarının doğru bir şekilde ele alınması sağlanmıştır. Sonuç olarak bu ödev, OOP prensiplerini, Java dilinin yeteneklerini ve temel algoritma tasarımını birleştirerek belirli kurallara göre çalışan bir simülasyon sistemi kurma becerisini geliştirmiştir. Karşılaşılan zorluklar, problem çözme ve hata ayıklama yeteneklerini pekiştirmiştir.