T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

G221210580 – MUSTAFA MASRI

SAKARYA

Nisan, 2025

Programlama Dillerinin Prensipleri Dersi

ÖDEV BAŞLIĞI (Uygun bir başlık bulun)

Mustafa Masri

G221210580 – 2/B

Özet

Bu rapor, nesne yönelimli prensiplere bağlı kalarak Java tabanlı bir uzay seyahati simülasyonunu detaylandırmaktadır. Problem, her biri yerel takvimleri etkileyen potansiyel olarak benzersiz gün uzunluklarına sahip gezegenler arasında yolcuların uzay gemileriyle hareketini simüle etmeyi içermekteydi. Temel gereksinimler arasında yolcu ömürlerini, gezegen tarihlerine dayalı gemi kalkışlarını, seyahat sürelerini ve karmaşık gemi imha koşullarını (transit sırasında veya varış sonrası yolcu ölümü) yönetmek yer alıyordu. Çözüm, varlıkları ve etkileşimleri modellemek için ayrı sınıflar (Planet, Spaceship, Person, Time, Simulation, FileReader) kullanır. Veriler metin dosyalarından okunur. Simulation sınıfı, saatlik ilerlemeyi yönetir, varlık durumlarını günceller ve biçimlendirilmiş konsol çıktısını görüntüler. Farklı gezegen zaman akışlarını ve incelikli imha mantığını (varıştan sonra bile orijinal yolcuları kontrol etme) ele almak merkezi zorluklardı. Program, gerekli uzay seyahati dinamiklerini başarıyla modelleyerek gezegenler arası simülasyonun dinamik bir görünümünü sunar.

Anahtar Kelimeler: Java, Nesne Yönelimli Programlama Simülasyon, Uzay Seyahati.

GELİŞTİRİLEN YAZILIM

Bu bölüm, ödevin gerektirdiği uzay seyahati simülasyon yazılımını geliştirmek için benimsenen yaklaşımı özetlemektedir. Birincil amaç, girdi dosyalarında tanımlanan parametrelere dayalı olarak gezegenler, uzay gemileri ve insanlar arasındaki etkileşimleri zaman içinde yansıtan dinamik bir simülasyon oluşturmaktı. Yazılım tamamen Java kullanılarak nesne yönelimli programlama (OOP) yaklaşımıyla geliştirilmiştir.

OOP tercihi, kendine özgü nitelikleri ve davranışları olan farklı varlıkları (gezegenler, gemiler, insanlar) içeren problemin doğasından kaynaklanmaktadır. Bu paradigma, modüler ve bakımı kolay bir tasarım sağladı. Her biri temel bir kavramı temsil eden geliştirilen ana sınıflar şunlardır:

Person: Yolcu verilerini (isim, yaş, ömür - saatlik olarak azalan, mevcut konum) kapsüller. Birincil sorumluluğu, bireysel ömrü ve durumu yönetmektir.
Time: Tüm tarih ve saat karmaşıklıklarını ele alır. Önemli bir özelliği, çeşitli gezegenler için farklı gün uzunluklarını yönetebilmesi, böylece planlanmış gemi kalkışları için kritik olan doğru tarih ilerlemesini ve karşılaştırmasını sağlamasıdır.
Planet: Bir konumu temsil eder. Yerel zamanını bir Time nesnesi aracılığıyla tutar, population listesini (Person nesnelerinin bir List'i) yönetir ve gemi varış/kalkışlarında insan ekleme ve çıkarma işlemlerini gerçekleştirir. Ayrıca her saat nüfus listesini ölen bireylerden temizler.
Spaceship: Taşıma araçlarını modeller. Yolculuğunu (kalkış/varış, tarihler, süre, kalan süre) takip eder ve mevcut passengers listesini yönetir. Ayrıca karmaşık imha mantığını ve görüntüleme gereksinimlerini

doğru bir şekilde uygulamak için başlangıçtaki yolcu listesini (originalPassengers) ve actualArrivalDate bilgisini saklar.
FileReader: Veri yükleme modülü olarak görev yapar. Gerekli .txt dosyalarını (Gezegenler.txt, Araclar.txt, Kisiler.txt) ayrıştırır, gerekli nesneleri oluşturur ve insanları başlangıç gezegenlerine yerleştirerek veya gemileriyle ilişkilendirerek ilk kurulumu gerçekleştirir. İlk kurulum sırasında verimli nesne aramaları için placePeople içinde dahili olarak HashMapler kullanılmıştır.
Simulation: Merkezi yönetici. Tüm varlıkların ana koleksiyonlarını (planets, ships, allPeople) tutar. start() metodu, saat saat ilerleyen çekirdek simülasyon döngüsünü içerir. Döngü içinde gerekli güncellemeleri yönlendirir: küresel ömür azalması, gezegen zaman/nüfus güncellemeleri, gemi durumu güncellemeleri (kalkış, transit, varış) ve simülasyonun bitiş koşulunu kontrol eder. Ayrıca her döngüde konsolu temizlemeyi ve biçimlendirilmiş çıktıyı görüntülemeyi yönetir.
Main: FileReader'ı başlatmaktan, Simulation örneğini yüklenen verilerle oluşturmaktan ve süreci başlatmaktan sorumlu basit giriş noktasıdır.

Simülasyon yinelemeli olarak ilerler. Her yineleme bir saati temsil eder. Döngü içindeki işlem sırası (önce küresel güncellemeler, sonra gezegen, sonra gemi) tutarlı bir durumu korumak için seçilmiştir. Gemi kalkışları, kalkış gezegeninin mevcut tarihinin (kendi Time nesnesi tarafından sağlanan) geminin planlanmış kalkış tarihiyle karşılaştırılmasıyla tetiklenir. Varışlar, yolcuların varış gezegeninin nüfus listesine aktarılmasını tetikler. Nüfus sayımı mantığı, transit halindeki yolcuların hiçbir gezegenin sayımına dahil edilmemesini doğru bir şekilde sağlar. İmha mantığı, hem transit sırasındaki ölümleri hem de tüm orijinal yolcuların simülasyon sona ermeden varış sonrası ölümünü kapsayacak şekilde dikkatlice uygulanmıştır.

Rapor yönergeleri uyarınca kod parçacıkları dahil edilmemiştir. Geliştirme, karmaşık algoritmalar yerine nesne etkileşimi yoluyla mantıksal doğruluğa ve simülasyon gereksinimlerini karşılamaya odaklanmıştır. Ödev muhtemelen OOP ilkelerini, dosya işlemeyi, simülasyon döngüsü tasarımını, durum yönetimini ve Java'da tarih/saat manipülasyonunu pekiştirmeyi amaçlamıştır. Başlıca zorluklar arasında tarih karşılaştırma mantığının doğru bir şekilde uygulanması (özellikle başlangıçtaki format uyuşmazlıkları ile) ve çok yönlü gemi imha koşullarının iyileştirilmesi yer almaktadır.

ÇIKTILAR

```
Simulation Hour: 635
Planets:
          --- X ---         --- Y ---         --- Z ---         --- V ---
Date        27.01.2025        02.02.2025        22.01.2025        06.02.2025
Population  7                 6                 0                 0

Spaceships:
Ship Name    Status       Departure  Destination Hours Remaining      Arrival Date
A1           Arrived      X          Y           0                   14.01.2025
A2           Destroyed    Y          V           --                  --
A3           Arrived      Z          X           0                   18.01.2025
B1           Destroyed    X          Z           --                  --
B2           Arrived      Z          V           0                   28.01.2025
C1           Arrived      V          X           0                   27.01.2025

Simulation complete after 635 hours.
```

SONUÇ

Bu proje, nesne yönelimli ilkeler kullanılarak gezegenler arası seyahatin işlevsel bir Java simülasyonunu başarıyla sunmuştur. Program, farklı gezegenlerde zaman ilerlemesini, yolcu yaşam döngülerini, planlanmış gemi hareketlerini ve gerekli imha mantığını doğru bir şekilde modelleyerek tüm temel ödev gereksinimlerini karşılamaktadır. Geliştirme süreci, dinamik bir simülasyonun karmaşıklığını yönetmede NYP'nin etkinliğini göstermiştir. Temel Java programlama kavramları, dosya işleme, durum yönetimi ve hata ayıklama konularında değerli pratik deneyim sağlamıştır. Kullanılan tasarım ilkeleri temel olup yazılım geliştirmede geniş ölçüde uygulanabilir.