



SAKARYA
ÜNİVERSİTESİ

Uzay Aracı Simülasyonu

1. Ödev - Java

2025-2026

Younes Rahebi | B221210588

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ

Dr.Öğr.Üyesi MUHAMMED FATİH ADAK

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

1. Giriş

Bu projede, gezegenler, uzay gemileri ve insanları içeren bir simülasyon geliştirdim. Projenin amacı, gezegenler arasındaki uzay gemisi hareketlerini, zaman yönetimini ve gezegenlerin nüfuslarını simüle etmektir. Sizden istenen temel gereksinimler arasında gezegenlerin zamanını ilerletmek, uzay gemilerinin hareketini yönetmek ve her gezegenin nüfusunu hesaplamak vardı.

2. Öğrendiklerim

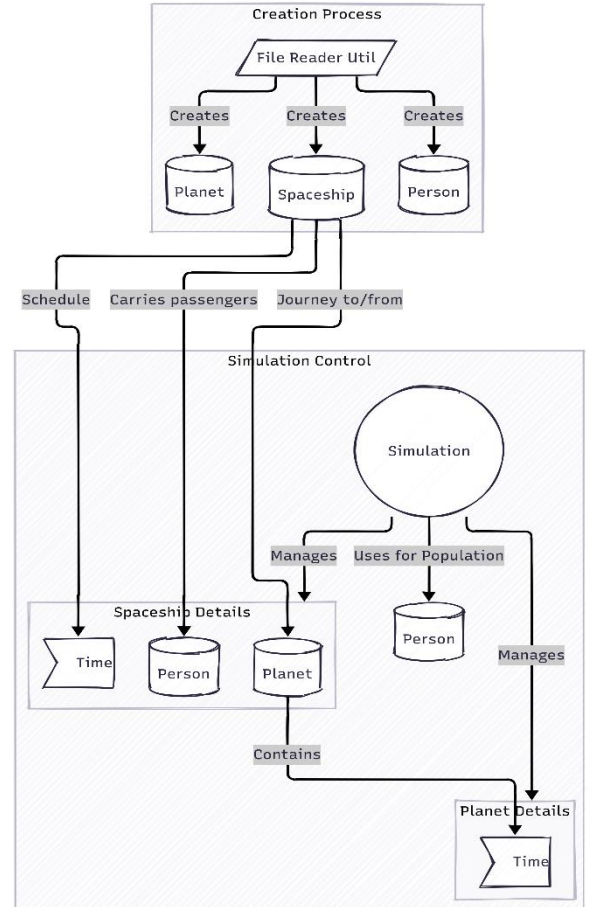
Bu projede birçok yeni bilgi ve beceri edindim:

- ❖ **Nesne Yönelimli Programlama:** Sınıflar, nesneler ve kalıtım gibi temel kavramları pekiştirdim.
- ❖ **Veri Yapıları:** Listeler ve hash map'ler gibi veri yapılarını kullanarak verimli veri yönetimi sağladım.
- ❖ **Dosya Giriş/Çıkış İşlemleri:** Dosyalardan veri okuma ve yazma işlemlerini öğrendim.
- ❖ **Simülasyonlarda Zaman Yönetimi:** Gezegenler için özel zaman sistemleri tasarladım ve zamanı ilerletme mantığını kavradım.
- ❖ **Java'da LocalDate Kullanımı:** Tarih yönetimi için LocalDate sınıfını kullanarak daha güvenilir ve standart bir yaklaşım benimsedim.

3. Ödevde Yaptıklarım

Projede aşağıdaki sınıfları oluşturdum ve işlevlerini gerçekleştirdim:

- ❖ **Planet:** Gezegenlerin adını, zamanını ve nüfusunu yönetir. Her gezegenin kendine özgü bir günlük saat sayısı (dayHour) vardır.
- ❖ **Spaceship:** Uzay gemilerinin hareketini, yolcularını ve durumunu (yolda, varıldı, imha) yönetir. Uzay gemilerinin kalkış ve varış tarihlerini hesaplar.
- ❖ **Person:** Bireyleri temsil eder ve yaşam sürelerini yönetir. Zaman ilerledikçe yaşam süreleri azalır.
- ❖ **Time:** Tarih ve zaman hesaplamalarını yapar. LocalDate kullanarak tarih yönetimini iyileştirdim.
- ❖ **FileReaderUtil:** Dosyalardan kişi, gezegen ve uzay gemisi verilerini okur ve ilgili nesneleri oluşturur.
- ❖ **Simülasyon:** Simülasyonun ana döngüsünü yönetir. Her saat başı gezegenlerin zamanını ilerletir, uzay gemilerinin durumunu günceller ve gezegenlerin nüfuslarını hesaplar. Ayrıca, simülasyonun durumunu konsola yazdırır.



4. Zorlandığım Kısımlar

Projede bazı zorluklarla karşılaştım:

- ❖ **Farklı Gezegenler Arasında Zaman Senkronizasyonu:** Gezegenlerin farklı günlük saat sayıları olduğundan, uzay gemilerinin kalkış ve varış zamanlarını senkronize etmek karmaşık hale geldi. Ancak, daha fazla test ve doğrulama yaparak bu sorunu çözdüm.
- ❖ **Nüfus Hesaplamalarının Doğruluğu:** Uzay gemilerinin durumlarına (yolda, varıldı, imha) bağlı olarak nüfus hesaplamalarında hata yapmamak için dikkatli olmam gerekti. Kenar durumlarını ele alarak ve test senaryoları geliştirerek doğru sonuçlar aldım.

5. Sonuç

Bu projede, nesne yönelimli programlama ve veri yapıları konusundaki bilgilerimi pekiştirdim. Simülasyonun temel gereksinimlerini başarıyla yerine getirdim ve optimizasyonlar yaparak performansı artırdım. Gelecekte, kullanıcı arayüzü ekleyerek simülasyonu görselleştirmeyi veya daha karmaşık senaryoları simüle etmeyi düşünebilirim. Ayrıca, hata işleme ve giriş doğrulamasını güçlendirerek projeyi daha sağlam hale getirebilirim.

```
C:\Windows\System32\cmd.e  x  +  v

Gezegenerler:

          ---X---          ---Y---          ---Z---          ---V---
Tarih    13.05.2025        08.02.3450        14.01.2651        10.08.1914
Nufus    3                0                0                2

Uzay Araclari:

Arac  Adi      Durum      Cikis      Varis      Hedefe Kalan Saat      Varis Tarihi
A      Vardi    X          V          0          09.08.1914
B      IMHA     Y          Z          --          ---
C      Vardi    Z          X          0          13.05.2025
D      IMHA     V          Z          --          ---
E      IMHA     X          Y          --          ---

TUM UZAY ARACLARININ HEDEFLERINE VARDI VEYA IMHA OLDU.
```