



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU**

**ÖDEV BAŞLIĞI**

**B231210098 - Hakan Efe Tüysüz**

**SAKARYA**

**Nisan, 2025**

Programlama Dillerinin Prensipleri Dersi

# GEZEĞENLER ARASI UZAY ARACI i UYGULAMASI

Hakan Efe Tüyüsüz

<sup>a</sup> B231210098 1. Öğretim C Grubu

## Özet

Bu ödevde, gezegenler arası seyahat simülasyonu ile ilgili bir problem ele alınmıştır. Amacımız, çeşitli araçların gezegenler arası hareketini ve zamanlamasını modellemektir. Her aracın bir çıkış gezegeni, varış gezegeni ve belirli bir çıkış tarihi bulunmaktadır. Bu araçların durumu, seyahatte olup olmadıkları ve hedefe kalan sürelerine göre sürekli olarak güncellenmektedir. Ayrıca, simülasyona dayalı olarak araçların hedeflerine ulaşma veya imha olma durumları da takip edilmektedir. Çözüm olarak, araçlar ve gezegenler arasındaki etkileşimi modellemek için bir harita yapısı (Map) kullanılmış, gezegenlerin zaman bilgileri ile araçların seyahat bilgileri karşılaştırılmıştır. İlgili koşullar kontrol edilerek, her aracın durumu doğru şekilde kullanıcıya bildirilmiştir. Bu çözümde veri yapıları ve algoritmalar kullanarak verimli bir simülasyon sağlanmıştır. NOT: Ödevde Thread.sleep(0); kullanılmıştır eğer istenirse bu değer ödev kontrolcüsü tarafından yükseltilebilir.

## © 2025 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

**Anahtar Kelimeler:** Seyahat simülasyonu, araçlar, gezegenler, zaman yönetimi

## 1. GELİŞTİRİLEN YAZILIM

Bu yazılım, bir uzay aracı simülasyonu gerçekleştirmek için tasarlanmıştır. Simülasyonun amacı, birden fazla gezegen arasında seyahat eden araçların ve bu araçların içerisindeki yolcuların zamanla olan etkileşimlerini modellemektir. Her araç, belirli bir gezegenden çıkış yaparak, başka bir gezegene varmaya çalışırken, çeşitli durumlar (örneğin araçların imha olması, yolcuların ömrünün tükenmesi, hedefe ulaşma) gözlemlenmektedir.

### 1.1. Kullanılan Yöntemler ve Yapılar

Yazılım, temel olarak şu veri yapıları ve algoritmalara dayanır:

- Listeler:** Kisi, Gezegen ve UzayAraci nesneleri gibi dinamik veri kümelerini saklamak için listeler kullanılmıştır. Bu sayede her gezegen, her araç ve her kişi için veri tutulması sağlanmıştır.
- Haritalar (Maps):** Map veri yapısı, araçlar ve gezegenler arasındaki ilişkileri (örneğin, araçların hangi gezegenden çıktığı, hangi gezegene varacağı, araçların hedefe ne kadar mesafesinin kaldığı gibi) yönetmek için kullanılmıştır. Bu yapı, hızlı erişim sağlayarak simülasyonun verimli çalışmasını sağlamaktadır.

- **Setler:** Hedefe ulaşmış ya da imha olmuş araçlar gibi benzersiz öğeler için Set veri yapısı tercih edilmiştir.

Yazılımın ana işleyişinde, gezegenler arası seyahat simülasyonu başlatıldıktan sonra her gezegenin zamanının bir saat ilerlemesi sağlanmaktadır. Bu ilerleme ile birlikte araçların seyahat süreleri ve yolcularının durumu güncellenmektedir. Aynı zamanda, araçların belirli bir gezegene varışları ve imha durumları da kontrol edilmektedir. İmha olan araçlar ve varış yapan araçlar ayrı bir şekilde izlenir.

## 1.2. Amaç ve Tasarım

Bu ödevin amacı, özellikle zaman ve durum güncellemelerini yönetebilecek bir simülasyon sistemi tasarlamaktır. Gezegenler arası seyahat eden araçların durumu ve yolcularının yaşlanma süreci, simülasyon boyunca sürekli izlenmiştir. Yazılım, araçların hedeflerine ulaşıp ulaşmadığını, araçların imha olup olmadığını ve gezegenlerdeki nüfusu güncel olarak takip etmektedir.

## 1.3. Zorluklar ve Çözüm Yöntemleri

- **Zorluk:** Aracın varış tarihi ile ilgili bir hata olması durumunda, aracın durumu doğru bir şekilde güncellenemiyordu.
- **Çözüm:** Bu hatayı gidermek için, aracın varış tarihi doğru bir şekilde kontrol edilerek yalnızca varış yapan araçlar için işlem yapılması sağlandı.

Ayrıca, simülasyonun doğru çalışabilmesi için her araç için kalan mesafe ve hedefe ulaşma durumu takip edilmiştir. Bu sayede her aracın ilerleyişi doğru şekilde izlenebilmiştir.

## 2. ÇIKTILAR

Geliştirilen simülasyon yazılımı, kullanıcıdan alınan gezegenler, uzay araçları ve kişiler gibi verilerle işlem yaparak her bir uzay aracının seyahat durumunu, varış tarihlerini ve gezegenlerdeki nüfusu hesaplar. Yazılımın çıktıları, simülasyonun doğru şekilde çalışıp çalışmadığını görmek adına oldukça önemlidir. Çıktıların doğru yorumlanabilmesi için yazılımın çeşitli girdilerle nasıl çalıştığını analiz edeceğiz.

### 1. Gezegen Durumları:

Her gezegenin, gezegenin adı, tarihi ve o gezegendeki nüfus sayısı gibi bilgileri içeren çıktılar üretiriz. Örneğin, bir gezegenin simülasyona dahil edilmesinin ardından, o gezegende yaşayan kişiler ve o gezegenin zaman dilimi göz önünde bulundurularak, bu gezegende hangi kişilerin yaşadığını ve hangi uzay aracının bu gezegenden kalmaya hazırlanıp hazırlanmadığını görmeliyiz. Bu bilgiler yazılımın her döngüsünde güncellenir.

#### Örnek Çıktı:

Gezegen: Mars

Tarih: 2025-04-26  
Nüfus: 150

## 2. Uzak Araçlarının Durumu:

Simülasyon her bir uzak aracının durumunu takip eder. Aracın durumu, üç temel kategoriye ayrılır:

- **Bekliyor:** Araç henüz çıkış yapmamış ve bekleme aşamasında.
- **Seyahatte:** Araç, çıkış gezegeninden hareket etmiş ancak henüz varış gezegenine ulaşmamıştır.
- **Varış Yapıldı:** Araç, hedef gezegenine ulaşmış ve yolculuk tamamlanmıştır.
- **İmha Edildi:** Araç, yolculuk sırasında tüm yolcularını kaybetmiş ve imha olmuştur.

Her bir araç için bu durumlar yazılımın her aşamasında güncellenir ve ekrana basılır.

### Örnek Çıktı:

--- Uzay Araçları Durumu ---

Arac: Apollo 1 | Durum: Seyahatte | Çıkış: Earth | Varış: Mars | Hedefe kalan saat: 12 | Varış tarihi: 06.10.2026

Arac: Voyager | Durum: Varış Yapıldı | Çıkış: Earth | Varış: Jupiter | Hedefe kalan saat: 0 | Varış tarihi: 04.10.2025

## 3. SONUÇ

Bu çalışmada geliştirilen yazılım, uzay yolculuklarında zaman dilimlerinin etkilerini simüle ederek, uzay araçlarındaki yolcuların yaşlanma süreçlerini ve seyahat sürelerini analiz etmektedir. Yazılımın temel amacı, farklı gezegenler arasındaki zaman farklarını hesaplayarak, uzay görevleri için daha doğru planlamalar yapar .

### Elde Edilen Sonuçlar:

- **Zaman Yönetimi:** Yazılım, gezegenler arası zaman farklarını doğru şekilde hesaplayarak seyahat sürelerini tahmin etmeyi sağlar.
- **Yaşlanma ve Biyolojik Etkiler:** Uzay yolculukları sırasında yolcuların biyolojik saatlerinin nasıl etkilendiği gözlemlenebilir.
- **Simülasyon Senaryoları:** Yazılım, farklı senaryoları test ederek uzay görevlerinde karşılaşılabilecek sorunları önceden görmeyi mümkün kılar.

### Gerçek Hayata Faydalar:

- **Uzay Görevleri Planlaması:** Yazılım, uzun süreli uzay yolculukları için daha güvenli ve sağlıklı görev planlamaları yapılmasına yardımcı olur.
- **Sağlık Takibi:** Astronotların sağlık durumları ve yaşlanma süreçleri daha iyi izlenebilir.

[1] ] T. M. Smith, "Introduction to Object-Oriented Programming with Java," 4th ed., Pearson, 2021.