

# SAKARYA ÜNİVERSİTESİ



**SAKARYA**  
ÜNİVERSİTESİ

## PROGRAMLAMA DİLLERİNİN PRENSİPLERİ

2024-2025 bahar dönemi  
Dr.Öğr.Üyesi MUHAMMED FATİH ADAK

**SELİM ALTIN**

**G231210558 2.A**

selim.altin@ogr.sakarya.edu.tr

### Ödev Raporu

**Gezegenler Arası Uzay Aracı Simülasyonu**

**2025**

**BİLGİSAYAR MÜHENDİSLİĞİ**

## Rapor Özeti

Bu raporda, “Gezegenler Arası Uzay Aracı Simülasyonu” projesinin temel gereksinimleri, uygulama sürecinde öğrendiklerim, karşılaştığım zorluklar ve çözümlerim özetlenmiştir. Projede, farklı gezegenlerde bağımsız zaman akışı, araç hareketi, yolcu ömrü yönetimi, imha ve varış koşulları eksiksiz olarak hayata geçirilmiştir. Simülasyon sonunda, araçların durumları net biçimde raporlanmış; gezegenlerin nüfus ve zaman bilgileri, araç özet istatistikleriyle birlikte sunulmuştur.

## 1. Projeden Beklenenler

- **Veri okuma:** Kisiler.txt, Gezegenler.txt ve Araclar.txt dosyalarından kişileri, gezegenleri ve uzay araçlarını parse etmek.
- **Zaman yönetimi:** Her gezegenin gün uzunluğuna uygun saat ve tarih simülasyonunu yürütmek.
- **Araç hareketi:** Çıkış tarihine ulaşıldığında araçları aktive etmek, mesafe azaldıkça hedefe ilerletmek.
- **Yolcu ömrü:** Saat başı yolcuların kalan ömrünü azaltmak; ölenleri sistemden silmek ve araçları imha etmek.

## 2. Projenin Başlangıcı ve Planlama

Nisan ayının ilk haftasında ödevi okumaya başladım, gereksinimleri madde madde not ettim ve Eclipse’te temel klasör yapısını kurdum. Önce model, io ve sim paketlerini oluşturarak işe giriş yaptım. Bu aşamada her bir sınıfın (Kisi, Gezegen, UzayAraci, DosyaOkuma, Simulasyon, Zaman) sorumluluğunu belirledim ve proje yol haritasını çıkardım: önce dosya okuma, sonra zaman yönetimi, en son simülasyon döngüsü ve raporlama.

## 3. Dosya Okuma

“Kisiler.txt”, “Gezegenler.txt” ve “Araclar.txt” dosyalarını okumak için önce DosyaOkuma sınıfını yazdım. Başlangıçta sabit format bekleyince başlık satırını atlamayan ve hatalı satırlarda çöken kod yaptım; denemelerde dosyalar farklı başlık içerince kayıplar yaşadım. Bunu, önce parse etmeye çalışıp başarısız olursa satırı atlayan bir yapıyla çözdüm. Veri parse metotlarını (parseKisi, parseGezegen, parseUzayAraci) modülerleştirdim.

## 4. Zaman Sınıfı ve 30 Günlük Ay Varsayımı

Gezegenlerin her birinin “gün kaç saatten oluştuğu” ve başlangıç tarihi verileriyle (“5.10.2020” gibi) çalışacak Zaman sınıfını oluşturdum. Ay uzunluklarını 30 gün olarak sabitledim; ödevde “aylık takvim” belirtilmediği için her ayı 30’er gün alarak tarih geçişlerini kodladım. Tarihleri “04.04.2025” formatına çevirme ve karşılaştırma işlemleri (compareTo) başta kafa karıştırdı, gün ve ayların başında sıfır dolgu yapmayı ihmal edince araçlar aktifleşmedi ve sonsuz döngüye girdim.

Zaman Karşılaştırma (ondanOnceMi Metodu)

Projenin ilerleyen aşamasında araçların çıkış tarihlerinin “geçmişte mi kaldığını” daha esnek kontrol edebilmek için Zaman sınıfına ondanOnceMi adında bir metot ekledim. Bu metot gün/ay/yıl değerlerini sırayla karşılaştırarak, mevcut gezegen zamanının verilen tarihten önce olup olmadığını döndürüyor. Böylece tarih tam tutmasa bile “tarih geçtikten sonra” da aracı aktive edebiliyoruz. Bu sayede tarih formatı farklarından veya saat atlamalarından kaynaklanan eksik aktivasyonlar ortadan kalktı, sonsuz döngü sorunu tamamen çözüldü.

## 5. Simülasyon Döngüsü ve Durum Yönetimi

Simülasyon sınıfında her saat başı:

1. Gezegen zamanlarını 1 saat ilerlettim,
2. Araçların çıkış tarihini kontrol edip aktive ettim,
3. Aktif araçlardaki yolcuların kalan ömrünü azalttım,
4. Ölenleri listeden çıkardım ve araç içi yolcu kalmazsa imha ettim,
5. Hedefe ulaşan araçları işaretleyip varış tarihini kaydettim,
6. Konsolda gezegen ve araç tablolarını hizalı printf ile yazdırdım.

Bu döngüyü bitirip son ekranı görmek çok tatmin ediciydi; en başta donan ve yavaşlayan sistem, performans iyileştirmeleri (Thread.sleep süresini kısaltma ve konsol temizlemeyi optimize etme) sonrasında akıcı çalıştı.

## 6. Karşılaşılan Zorluklar

- **Başlık satırı kontrolü:** Txt dosyalarında başlık olup olmadığını otomatik algılamak zorlayıcıydı.
- **Tarih formatlama:** “4.4.2025” vs. “04.04.2025” çelişkileri sonsuz döngüye neden oldu.
- **Ay uzunluğu kararı:** Ayları 30 gün kabul etmek mantıklı ama gerçekçi değil; burada varsayıma sadık kaldım.
- **Iterator ile silme:** Yolcu silerken ConcurrentModificationException hatası aldım, iterator kullanarak çözdüm.
- **Sonsuz döngü:** Araçlar çıkış tarihine hiç ulaşmadıkları için döngü bitmiyordu → ondanOnceMi() ile geçmiş tarihe de izin verildi.
- **Geniş veri seti:** Yüzlerce yolcu ve araçla test ederken bellek ve hız sorunları çıktı; kodu sadeleştirip gereksiz tekrarları kaldırdım.
- **Konsol temizleme:** Windows ve Unix farklı escape kodları kullanınca, platform uyumsuzlukları ortaya çıktı.
- **Modülerlik ihtiyacı:** Başlangıçta tüm işlem tek yerdeydi, okunması zordu → her aşama ayrı metoda taşındı.

## 7. Eksikler / İyileştirmeler:

Başlangıçta gezegen nüfusunu dinamik hesaplamak için baslangictaNufusEkle() fonksiyonu yazıldı, ancak karmaşık senaryolarda nüfus dengesi daha da test edilebilir..

## 8. Programda Kullandığım Veri Yapıları ve Teknolojiler

- ArrayList<Kisi>, ArrayList<Gezegen>, ArrayList<UzayAraci> → tüm nesne listeleri; dinamik ekleme/çıkarma için.

- `HashMap<String, Zaman>` → gezegen adı → zaman simülasyonu eşlemesi; farklı zaman akışlarını yönetmek için.
- `Iterator<Kisi>` — ömür sıfırlanan kişileri listeden güvenli şekilde silmek için.
- `Thread.sleep(0)` — döngü hız kontrolü.
- Paketsel yapı (model, io, sim) — OOP prensiplerine uygun sorumluluk ayrımı.

## 9. Sonuç ve Öğrendiklerim

Bu proje sayesinde gerçek dünya verisi okuma, tarih–zaman yönetimi, iterator ile güvenli koleksiyon işlemleri, modüler tasarım ve performans optimizasyonu konularında pratik yaptım. Sonsuz döngü, tarih uyumsuzluğu ve dosya formatı esnekliği gibi zorluklar öğrenme sürecimi hızlandırdı. Sonunda hem ödev gereksinimlerini karşıladım hem de kod kalitesini artıracak alışkanlıklar edindim.

## 10. Çıktı

```

--- Simülasyon Bitti ---
Gezegenler:

--- X ---          --- Y ---          --- Z ---          --- W ---          --- V ---
Tarih:             03.05.2025         01.05.2025         04.05.2025         06.05.2025         02.05.2025
Nüfus:             4                  1                  2                  0                  2

Uzay Araçları:
Araç Adı  Durum   Çıkış   Varış   Hedefe Kalan Saat   Hedefe Varacağı Tarih
A         Vardı   X       Y       0                 26.4.2025 - 0:00
B         Vardı   Y       Z       0                 27.4.2025 - 14:00
C         İMHA   Z       W       --                --
D         Vardı   W       V       0                 26.4.2025 - 19:00
E         Vardı   V       X       0                 29.4.2025 - 17:00
F         İMHA   X       Z       --                --
G         Vardı   Y       V       0                 27.4.2025 - 2:00
H         Vardı   Z       X       0                 27.4.2025 - 3:00
I         İMHA   W       Y       --                --
J         İMHA   V       W       --                --

=== 📋 Simülasyon Özeti ===

Toplam geçen süre:      217 saat
Toplam başlangıç yolcusu: 20
Hayatta kalan yolcu:   9
İmha olan araç sayısı: 4
Ulağan araç sayısı:    6

=== 🕒 Gezegenlerin Son Zamanı ===
V: 2.5.2025 - 1:00
W: 6.5.2025 - 1:00
X: 3.5.2025 - 1:00
Y: 1.5.2025 - 7:00
Z: 4.5.2025 - 17:00

```