



**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Programlama Dillerinin Prensipleri Ödev Raporu

**Uzay Yolculuğu Simülasyon Programı**

**B231210071 - Oğuzhan Öztürk**

**Sakarya**

**Nisan, 2025**

Programlama Dillerinin Prensipleri

# Uzay Yolculuğu Simülasyon Programı

Oğuzhan Öztürk - B231210071 - 1C

## Özet

Uzay yolculuğu simülasyon programı, bu programda txt dosyalarından okuduğumuz uzay araçları, kişiler ve gezegenlerin Java dilinde nesneye dayalı programlamayla gerçekleştirilmesi istenmiştir. Bu programda uzay araçları iki farklı gezegen arasında seyahat yapacaktır. Bu esnada gezegenlerdeki farklı gün uzunlukları, farklı tarihler, varış süreleri, yolculukta kalan süre, araçlardaki insanların yaşam süreleri hesaba katılarak kullanıcıya istenilen bilgiler konsol aracılığıyla gösterilmelidir.

## 1. DOSYALARIN OKUNMASI

Dosya okuma işlemi için oluşturulan sınıfta `BufferedReader` ve `FileReader` sınıflarından faydalanılmıştır. Dosyaların konumu dist klasöründe, .jar dosyasıyla aynı yerde, olduğundan dolayı dosyalara erişmek için `System.getProperty("user.dir") + dosya_adi` ile erişilir ve okunan satırlar döngüler yardımıyla `Constructor`'lara verilip istenilen sınıfların nesneleri oluşturulur. Bu nesneler bir listeye eklenir ve dosya okuma işleminden bu liste döndürülür. Böylelikle dosyalardaki bütün Gezegen, Araç ve Kişiler artık programda nesne halinde listelerde bulunur.

## 2. SİMÜLASYON

Simülasyonda ilk iş dosyaların okunup listelere kaydedilmesidir. Sonra kullanıcıdan simülasyondaki 1 saatin gerçekte kaç milisaniyede geçmesini istediğini sorar ve kaydeder. While döngüsüne girmeden önce bir sayaç değişkeni oluşturulur bu değişken programda toplam kaç saatlin simüle edildiğini sonda göstermek için vardır. While döngüsünün şartı simülasyonun bitip bitmediğini kontrol eden fonksiyonun değeridir.

Bu fonksiyona araç ve kişi listeleri parametre olarak verilir. Fonksiyonda araçların durumu for-each döngüsü (Java'da for döngüsünün geliştirilmiş hali) ile kontrol edilir döngüdeki aracın durumu "Yolda" ya da "Bekliyor" ise for-each döngüsü devam ettirilmeden false döndürülür ve simülasyonun devam edeceği kabul edilir. Eğer listedeki hiçbir aracın durumu ikisinden biri değilse true döndürülür ve simülasyonun biteceği kabul edilir.

While döngüsü içinde konsolda çıktılar arası karışıklık olmasın diye for döngüsü ile 70 satır boşluk bırakılır. Araç, kişi, gezegen listeleri zaman sınıfının bir saat ilerletme fonksiyonları yardımıyla güncellenir. Listeler aynı zamanda bu fonksiyonlara parametre olarak verilmiştir.

Gezegenler:					
	---A0---	---B0---	---C0---	---D0---	---E0---
Tarih	31.08.2025	29.04.2025	21.12.2025	08.02.2027	22.09.2024
Nufus	2	0	2	2	1
Uzay Araclari:					
Arac	Adi	Durum	Cikis	Varis	Hedefe Kalan Saat
Arac 00	Imha	A0	B0	---	---
Arac 01	Imha	A0	B0	---	---
Arac 02	Vardi	B0	A0	0	02.06.2025
Arac 03	Vardi	B0	A0	0	31.08.2025
Arac 04	Imha	C0	A0	---	---
Arac 05	Imha	A0	C0	---	---
Arac 06	Vardi	D0	C0	0	27.05.2025
Arac 07	Vardi	A0	D0	0	01.05.2026
Arac 08	Imha	E0	B0	---	---
Arac 09	Vardi	C0	E0	0	20.05.2024
Simulasyon Bitti					
Geçen süre : 2350					

Şekil 1: Örnek txt dosyaları için simülasyon bitince ekrandaki değerler

Ekrana yazma fonksiyonu çağrılır bu fonksiyonda veriler istenilen formatta konsolda gösterilir(Şekil 1). Başta kullanıcıdan istenen bir döngüde kaç milisaniye bekleneneğini tutan hız değişkeni Thread.Sleep(int) fonksiyonuna parametre olarak verilip uygulamayı o kadar milisaniye bekletir. Sonrasında sayaç 1 arttırılır ve döngü başa döner ve tekrar kontrole girer eğer bu sefer döngü bitmiş ise ekrana geçen süre sayaç yardımıyla verilir.

### 3. ZAMAN

Zaman sınıfının 3 tane statik fonksiyonu vardır bu fonksiyonların üçü de parametre olarak araç, gezegen ve kişi listelerini parametre olarak alır ve bunlardan birinin tüm nesnelerinin zamanını 1 saat ilerletip geri döndürür.

Araç sınıfının zamanını ilerletirken önce aracın durumu güncellenir sonra araç sınıfına ait olan ilerletme fonksiyonu çağrılır. Bu fonksiyonda geçen toplam saat arttırıldıktan sonra varış ve çıkış gezegenlerinin de zamanı 1 saat ilerletilir eğer aracın durumu "Yolda" ise varmaya kalan saati tutan değişken de 1 azaltılır.

Gezegen sınıfının zamanı ilerletilirken sınıftaki geçen toplam saat değişkeni arttırılır ve eğer geçen toplam saat değişkeni gezegendeki 1 günün saatiyle bölümü kalansız ise gezegendeki tarih 1 arttırılır.

Kişi sınıfının zamanı ilerletilirken geçen toplam saat arttırılır ve eğer kişi yaşıyorsa ömründe kalan saatler de 1 azaltılır.

## ARAÇLARIN DURUMLARI

Araçların durumlarının doğru hesaplanması simülasyondaki en önemli kısımlardandır simülasyonun bitip bitmediğini kontrol ederken ve gezegendeki nüfusu hesaplarken kontrol edilir.

- *Vardı durumu:* Eğer araçta varış gezegenindeki tarih Constructor'da mesafe ve çıkış tarihi yardımıyla hesaplanan varış tarihinden sonra veya eşitse ve kalan mesafenin saat olarak tutulduğu değişken 0 ise aracın durumu Vardı kabul edilir.
- *Bekliyor durumu:* Eğer çıkış gezegenindeki tarih çıkış tarihinden önceyse araç bekleme durumundadır.
- *Yolda durumu:* Eğer çıkış gezegenindeki tarih çıkış tarihinden sonra veya eşit ise araç yola çıkmıştır ve yolda durumundadır.
- *İmha durumu:* Eğer araçtaki yaşayan kişi sayısı 0 ise diğer durumlar farketmeksizin araç imha edilir

## SONUÇ

Bu ödevde Java dilinde dosya okuma, uygulamayı beklemeye alma, tarih hesaplama, tarih karşılaştırma, zaman ilerletme gibi farklı işlemlerin nasıl yapıldığını anlamaya çalıştım. En çok zorlandığım kısım araç, kişi ve gezegenlerin zamanını birbirleriyle tutarlı ilerletmek ve tarihlerin ve kalan mesafelerin karşılaştırılmasıyla aracın durumunun belirlenmesini ayarlamak oldu.