



**T.C.
SAKARYA ÜNİVERSİTESİ**

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

PDP 1. Ödev

B221210003 – Akif Emre YAMAN

SAKARYA

Nisan, 2025

Programlama Dillerinin Prensipleri Dersi

ÖZET:

Projede insanlar uzay aracı ile bir gezegenden başka bir gezegene seyahat edecek ve gezegenlerin nüfusu olacak.

Uzay araçlarında belli sayıda kişiler olacak. Bu kişilerin belli bir ömrü olacak ve ömrü bitenler ölecek.

Uzay aracındaki herkes ölürse o uzay aracı İMHA olacak.

Uzay aracı bulunduğu gezegende kendi kalkış tarihi ile gezegenin bulunduğu tarih aynı olduğunda başka gezegene gidecek

Giderken yolda geçen zaman gideceği ülkenin saat dilimine göre hesaplanacak.

Ancak her gezegende zamanın akış hızı farklı olacak(mesela dünyada bir gün 24 saatken Satürn gezegeninde bir gün 10.7 saat sürüyor.)

Projemi yaparken bu ana şablon üzerinden ilerledim. Bazı ufak detaylar eklemeye çalıştım ör: Ayların hepsini sabit almadım dünya ile birebir tarih akışı sağladım artık yıldaki şubat ayının 29 gün olmasına kadar.

GELİŞTİRİLEN YAZILIM:

Geliştirdiğim yazılım nesne yönelimli bir yazılımdır ve genel olarak basit ve anlaşılır bir bellek organizasyonu yapmak istedim. Bizden beklenen minimum classlara projemde yer verdim

Gezegen, Uzayaracı, Kisi, Zaman, Simulasyon, DosyaOkuma ve Main(Simulasyon sınıfını çalıştırıyor).

Her sınıfın kendi sorumluluğu var ve kendi verileri ile davranışlarını kapsüllüyor.

Veriler data klasöründen DosyaOkuma classı ile okunarak nesnelere dönüştürülüyor. Yani DosyaOkuma classı veri okuma merkezi oluyor.

Zaman yönetimi, zaman classı ile yapılıyor. Dünya ile birebir takvim sistemi kullanarak her gezegenin farklı gün uzunluğunu destekliyor.

Uzay araçları ile ilgili kısmı UzayAracı classı yapıyor. Yolcu taşıma ve seyahat durumlarını yönetiyor. (Araç + "Bekliyor", "Yolda", "Vardı" ve "İMHA") Yolcu listelerini, varış süresini hesaplıyor.

Simülasyon classı ise projenin kontrol yeridir. Nüfus ve araç durumlarını sürekli günceller. Konsol arayüzü ona bağlıdır ve sonuçları sonuc.txt oluşturup oraya yazar.

ÇIKTILAR:

Bu yazılım istenildiği gibi esnek olarak geliştirilmiştir. Yeni şeyler eklemeye müsait olması hata çıkmasına neden olabilir. Hata çıkmaması için hata mesajları eklenmiştir. Ör: Null kontrolleri ile geçersiz durum kontrolleri yapılmata ve veri bütünlüğü için gettler/settler kullanılmıştır.

SONUC:

Bu proje bana karmaşık ve anlaması güç sorunları nasıl çözebileceğimi öğretti. Elimdeki kaynakları doğru kullanmanın önemini gördüm. Veri yapılar ve bellek organizasyonu gibi bildiğim konuları pekiştirerek java diline daha hakim oldum.

REFERANSLAR:

[1] Java Tutorial.