



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

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

**Java dili**  
**Rastgele Savaşan Koloniler**

**B211210031 - Eren KARA**

**SAKARYA**

**Mayıs, 2023**

Programlama Dillerinin Prensipleri Dersi

# Java Dili

## Rastgele Savaşan Koloniler

Eren Kara

Numara: B211210031 Grup: 1.C

### Özet

Bizden C-dilinde verilen nesne yönelimli benzetim ödevinin java dilinde tekrar yazılması istendi. Aradaki farkları karşılaştırmamız açısından oldukça iyi bir yöntem.

Ödevde rastgele şekilde savaşacak koloniler oluşturmamız istendi. Bu kolonilerin popülasyon ve yemek stoğu değerleri olacaktı. Bu popülasyon değeri kullanıcıdan alınacaktı ve yemek stoğu popülasyonun 2 katı kadar olacaktı. Bu 2 değerden herhangi biri 0'ın altına düştüğünde ise koloninin yaşamı sona erecekti.

Koloniler birbirleriyle savaşacak, savaşın sonunda kaybeden koloni popülasyon ve yemek kaybedecekti. Kazanan ise sadece yemek kazanacaktı. Bu savaşlar 1 koloni kalıncaya dek devam edecekti.

© 2023 Sakarya Üniversitesi.

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

Anahtar Kelimeler:Java, Rastgele Şekilde Savaşan Koloniler, Modülerite, Nesneye Dayalı Programlama, Programlama Dillerinin Prensipleri

## 1. GELİŞTİRİLEN YAZILIM

Odevde modüleriteye oldukça dikkat etmek istedim ki zaten bizden de istenen oydu. Bunun için birçok yapıyı direkt olarak kullanmaktansa yapılar arasına arayüzler koydum böylece birden fazla farklı sınıfın kullanılmasının önünü açtım. Örneğin koloni sınıfı ITopluluk arayüzünden kalıtım alıyor. ITopluluk sınıfı belirli başlı fonksiyonların implemente edilmesi için kontrat sağlıyor. Koloni ise bunları implemente ederek IToplulugun kullanılabilir bir halini oluşturuyor. Koloni sınıfı bu fonksiyonları hocamızın bizden istediği şekilde implemente ediyor. Örneğin savaş kaybeden koloni %10 yemek kaybediyor. Ancak ben ileride bu fonksiyonların mantığının değişmesini isteyebilirim. Bunun için tek yapmam gereken ITopluluk sınıfını gerçekleyen başka bir topluluk sınıfı yazmam ve ITopluluğu kullandığım yerlere yeni oluşturmuş olduğum bu topluluk sınıfını atamam. Böylece modüleritenin önünü açmış oluyorum. Bunu birçok sınıf için yaptım hatta bütün sınıflar birbirini kullanmak yerine birbirlerinin arayüzlerini kullanıyor diyebilirim.

### 1.1. Taktik ve Üretim Sınıfları

#### 1.1.1. Taktik Sınıfları

ITaktik sınıfını gerçekleştiriyorlar. 1 fonksiyona sahipler ve bu fonksiyon geriye savaş gücü döndürüyor.

Taktik sınıflarını oluştururken kaynakça[0]'dan esinlenerek oluşturmaya çalıştım ve savaş gücü döndürme mantıklarını buradaki anlatılanlara göre yaptım.

#### 1.1.2. Üretim Sınıfları

IÜretim sınıfını gerçekleştiriyorlar. 1 fonksiyona sahipler ve bu fonksiyon geriye tur sonu üretilecek olan yemek miktarını döndürüyor.

Üretim sınıflarını oluştururken, insanların yemek kaynaklarından ilham alarak oluşturdum. Geriye döndürdükleri yemek miktarı da bu üretim miktarlarının ne kadar yemek üretebilme potansiyelleri varsa ona göre oluşturuldu.

## **1.2. Topluluk Sınıfları**

ITopluluk arayüzünden kalıtım alan sınıflardır. Benim ödevimde sadece bir sınıf gerçeklemesi var. O da koloni sınıfı. Hocanın bizden istediği mantıkta fonksiyonlarını gerçekleştiriyor.

### **1.2.1. Koloni Sınıfı**

Koloni sınıfı ATopluluk abstract sınıfından kalıtım alıyor. ATopluluk ise ITopluluktan kalıtım alıyor. Koloni sınıfı içerisinde popülasyon, yemek stoğu, sembol, kazanma, kaybetme, taktik ve üretim fieldları bulunuyor. Yemek kaybı, yemek kazancı, savaş can kaybı, savaş, yemek üret gibi fonksiyonları bulunuyor. Bu fonksiyonların gerçekleştirilmesi hocamızın bizden istediği gibi. Ayrıca her bir koloninin sembolü o koloni için unique olma özelliği taşıyor ki bunu equals metodunu override etmede kullandım.

## **1.3. İçerik Sınıfları**

Kullanıcıdan içeriğin yani sayı dizisinin okunmasında yardımcı olan sınıflardır.

### **1.3.1. İçerik Kontrol Sınıfı**

Fonksiyonlarına gönderilen parametredeki string ifadenin sayılara ayrılabilip ayrılamadığını, hepsinin boşluktan oluşup oluşmadığını kontrol eden sınıftır. Fonksiyonlarının hepsi statik'tir.

### **1.3.2. İçerikParcala Sınıfı**

Fonksiyonlarına gönderilen string ifadeyi istenilen şekillere parçalayıp geri döndüren sınıftır. Eğerki parametre olarak gönderilen değer istenilen şekle parçalanamıyorsa ki bunu IcerikKontrol sınıfı sayesinde kontrol ediyor IOException hatası fırlatıyor. Bütün fonksiyonları statik'tir.

### **1.3.3. İçerik Sınıfı**

İçerisinde sayı dizisi okumaya yarayan bir fonksiyon ve bir tane de bu okunan değeri tutacak olan field bulunmakta. Fonksiyonu ile okuduğu sayı dizisini geri döndürmekte. Bu okuma işlemini yaparken IcerikParcala sınıfından yardım almakta.

## **1.4. Ekran Sınıfı**

IEkran arayüzü gerçekleyen sınıftır. Bu sınıf oyunun hangi akışa çıktığı vereceğini belirler. Ekran sınıfı bunu System.in ile System.out'u kullanarak gerçekler.

## **1.5. Olustur Sınıfı**

Olustur sınıfı IOlustur arayüzünü gerçekleştiriyor. İçerisinde toplulukları oluşturma mantığını barındırıyor. Ancak ben de bir tane topluluk ( koloni ) olduğundan sadece koloni oluşturup liste şeklinde geri döndürebiliyor.

Oluşturma işlemi sırasında kolonilere rastgele bir şekilde taktik ve üretim atıyor. Ayrıca kolonilere diğer kolonilerden farklı olacak şekilde bir sembol atama işlemi gerçekleştiriyor.

## **1.6. Oyun Mantığı Sınıfları**

### **1.6.1. Savaş Sınıfı**

ISavaş arayüzünü gerçekleştiriyor. İçerisinde savaş mantığını barındırıyor. Kendisine verilen eşleşmiş 2 tane topluluğu savaştırıp kaybedeni geri döndürüyor. Savaşın sonuçlarının uygulanmasında bu sınıfta gerçekleştiriyor.

### **1.6.2. Oyun Turu Sınıfı**

İçerisinde tur mantığını barındıran sınıftır. IOyunTuru arayüzünü gerçekler.

Tur fonksiyonu içerisinde Oyun sınıfından aldığı yaşayan koloniler listesindeki koloniler arasından bir eşleşme yapar. Eşleşmelerin tur içerisinde tekrarlanmaması için field olarak bu eşleşmeleri içerisinde tutar. Eslesme fonksiyonu tur içerisinde daha önce eşleşmemiş iki sınıfı birbiriyle eşleştirir ve geri döndürür. Tur fonksiyonu bu

```

<terminated> Test [Java Application] C:\Users\Eren\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Bir sayı dizisi giriniz:
20 30          50 10

-----

Tur sayisi: 0
Koloni  Populasyon  YemekStogu  Kazanma  Kaybetme
ê       20         400        0         0
š       30         900        0         0
ı       50        2500        0         0
É       10         100        0         0

-----

Tur sayisi: 1
Koloni  Populasyon  YemekStogu  Kazanma  Kaybetme
ê       21         580         2         1
š       22         754         1         2
ı       6         2094         1         2
É       --         --         --         --

```

### 3. SONUÇ

Ödevde C’de yazdığım kodun Java’da gerçekleştirmesinin ne derece rahat olduğunu farkettim. Dilin sunduğu NDP paradigmasının özelliklerinin kod yazımını hızlandırdığını, kodun okunabilirliğini arttırdığını, arka plandaki karmaşıklıktan bizleri soyutladığını anladım. Özellikle sınıflarla beraber gelen hazır fonksiyonların, veri yapılarının kullanımı kesinlikle kodu okumayı ve yazmayı inanılmaz basitleştiriyor. Bir dizinin boyutunu parametre olarak vermemek, bunu düşünmemek, dizi.length demek bu kadar değerliymiş demek diyor insan.

### Referanslar

- [1] <https://10layn.com/tarih-boyunca-uygulanmis-en-etkili-savas-taktikleri/>
- [2] <https://regex101.com/>
- [3] <https://www.geeksforgeeks.org/program-to-convert-array-to-list-in-java/>
- [4] <https://www.geeksforgeeks.org/arrays-stream-method-in-java/>
- [5] <https://stackoverflow.com/questions/8811815/is-it-possible-to-assign-numeric-value-to-an-enum-in-java>
- [6] <https://stackoverflow.com/questions/14319732/how-to-copy-a-java-util-list-into-another-java-util-list>
- [7] <https://stackoverflow.com/questions/3110547/java-how-to-create-new-entry-key-value>
- [8] <https://www.youtube.com/watch?v=nCvipKIDndM>
- [9] [https://www.w3schools.com/java/java\\_hashmap.asp](https://www.w3schools.com/java/java_hashmap.asp)