T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

ÖDEV BAŞLIĞI

SAKARYA

Mayıs, 2023

Programlama Dillerinin Prensipleri Dersi

KOLONİ SAVAŞ OYUNU

Selcan NARİN

G191210027 1B

SINIFLAR VE GÖREVLERİ:

Taktik.java
Soyut sınıf olan Taktik sınıfında **Savas** adlı soyut bir metot tanımlanmıştır.

ATaktik.java
Taktik sınıfından türemiş bir alt sınıftır. Savas fonksiyonu override edilir ve 0 ile 500 arasında bir değer döndürür. Bu değer savasGucu olarak Koloni sınıfında iki koloni savaşırken kullanılır.

BTaktik.java
Taktik sınıfından türemiş bir alt sınıftır. 500 ile 1000 arasında bir değer döndürür. Bu değer savasGucu olarak Koloni sınıfında iki koloni savaşırken kullanılır.

Uretim.java
Soyut sınıf olan Uretim sınıfında **Uret** adlı soyut bir metot tanımlanmıştır.

AUretim.java
Uretim sınıfından türetilmiş bir alt sınıftır. Uret fonksiyonu override edilir ve 0 ile 5 arasında bir değer döndürür. Bu değer sonrasında yemekStogu olarak Koloni sınıfında kullanılır.

BUretim.java
Uretim sınıfından türetilmiş bir alt sınıftır. Uret fonksiyonu override edilir ve 5 ile 10 arasında bir değer döndürür. Bu değer sonrasında yemekStogu olarak Koloni sınıfında kullanılır.

Koloni.java
- sembol: Kolooninin sembolünü temsil eden bir karakter.
- populasyon: Kolooninin nüfusunu temsil eden bir sayı.
- yemekStogu: Kolooninin yemek stoğunu temsil eden bir sayı.
- kazanma: Kolooninin kazandığı savaş sayısını temsil eden bir sayı.
- kaybetme: Kolooninin kaybettiği savaş sayısını temsil eden bir sayı.
- taktik: Kolooninin savaş stratejisini temsil eden bir Taktik nesnesi.
- uretim: Kolooninin üretim stratejisini temsil eden bir Uretim nesnesi.

savasYap metodu, iki koloni arasında savaşı yapar ve sonuçları belirler. İşleyişi aşağıdaki gibidir:

1) İlk olarak, savasYap metoduna iki koloni (koloni ve rakipKoloni) parametre olarak verilir.
2) savasGucu1 ve savasGucu2 değişkenleri oluşturulur ve her bir kolooninin taktik nesnesi üzerinden Savas() metodunu çağırarak rastgele bir savaş gücü elde edilir.
3) kazananKoloni ve kaybedenKoloni adında iki Koloni nesnesi oluşturulur.
4) Savaş güçleri karşılaştırılır ve kazanan koloni belirlenir:
5) Eğer savasGucu1 büyük ise, mevcut koloni (koloni) savaşı kazanır ve kazananKoloni değişkenine atanır, rakip koloni (rakipKoloni) ise kaybedendir ve kaybedenKoloni değişkenine atanır.
Eğer savasGucu1 küçük ise, rakip koloni (rakipKoloni) savaşı kazanır ve kazananKoloni değişkenine atanır, mevcut koloni (koloni) ise kaybedendir ve kaybedenKoloni değişkenine atanır.
Eğer savasGucu1 ve savasGucu2 eşit ise, popülasyonlara bakılır:
Eğer mevcut kolooninin popülasyonu (populasyon) rakip kolooninin popülasyonundan (rakipKoloni.populasyon) büyük ise, mevcut koloni kazanır.
Eğer mevcut kolooninin popülasyonu (populasyon) rakip kolooninin popülasyonundan (rakipKoloni.populasyon) küçük ise, rakip koloni kazanır.
Eğer popülasyonlar eşit ise, rastgele bir seçim yapılır ve bir koloni kazanır.

Kaybeden kolonin popülasyonu kontrol edilir. Eğer popülasyon 5 veya daha küçükse ve kazanan kolonin popülasyonu en az 1 ise, kaybeden kolonin popülasyonu 0 olarak ayarlanır. Oyunun sonucuna daha hızlı ulaşmak için böyle bir şart eklenildi.
6) Yemek stoğu transferi gerçekleştirilir:
7) Kaybeden kolonin yemek stoğu azalma oranına göre azaltılır.
Transfer edilen yemek miktarı, kaybeden kolonin yemek stoğunun azalma oranına göre hesaplanır ve kazanan kolonin yemek stoğuna eklenir.
Kazanma ve kaybetme sayıları güncellenir:
8) Kazanan kolonin kazanma sayısı bir artırılır.
Kaybeden kolonin kaybetme sayısı bir artırılır.

Uret fonksiyonu kolonin üretim stratejisine göre yemek üretir ve yemek stoğunu günceller.

Oyun.java

Oyun sınıfı, koloniler arasındaki savaş oyununun simülasyonunu gerçekleştirmek için kullanılır. Sınıf aşağıdaki özelliklere sahiptir:

- koloniler: Koloni sınıfından oluşan bir liste. Bu liste, oyunda yer alan tüm kolonileri içerir.
- turSayisi: Oyunun hangi turda olduğunu belirtir.
- hayattaKalanKoloniSayisi: Oyunda hayatta kalan koloni sayısını takip eder.
- Aşağıda Oyun sınıfının yöntemlerinin açıklamaları bulunmaktadır:

Oyun(List<Koloni> koloniler): Oyunun başlatılması için koloniler listesini alarak bir Oyun nesnesi oluşturur. turSayisi ve hayattaKalanKoloniSayisi başlangıçta sıfıra ayarlanır.

baslat(): Oyunun başlamasını sağlayan yöntem. Oyunun her turunda koloniler arasındaki savaşlar gerçekleştirilir ve sonuçlar ekrana yazdırılır. Savaşlar ve sonuçlar için döngüler kullanılır.

1) İlk olarak, başlangıç durumu ekrana yazdırılır. Her kolonin sembolü, popülasyonu, yemek stoğu, kazanma ve kaybetme sayıları görüntülenir.
2) Sonra, koloniKaldimi adlı bir boolean değişkeni true olarak ayarlanır. Bu değişken, oyunda hayatta kalan kolonin olup olmadığını belirlemek için kullanılır.
3) Döngü içinde, her turda savaşlar gerçekleştirilir ve kolonilerin durumu güncellenir.
4) İç içe döngüler kullanılarak her koloni diğer kolonilerle savaşır. Ancak hayatta olmayan veya yemek stoğu biten koloniler savaşmaz.
5)Savaş sonucunda savasYap yöntemi kullanılarak kazanan ve kaybeden koloniler belirlenir. Ardından uret yöntemi çağrılarak her turda üretim yapılır.
6)Her kolonin popülasyonu %20 oranında artar ve yemek stoğu, güncel popülasyonun iki katı kadar azalır.
7)Sonuçlar ekrana yazdırılır. Hayatta olmayan kolonilerin durumu belirli bir formatta görüntülenirken, hayatta olan kolonilerin sembolü, popülasyonu, yemek stoğu, kazanma ve kaybetme sayıları görüntülenir.
8)hayattaKalanKoloniSayisi kontrol edilerek oyunda sadece bir koloni hayatta kalmışsa döngü sonlandırılır ve oyun biter.

Test.java

Kullanıcıdan alınan koloni popülasyonlarıyla bir koloni savaşı oyunu yapmayı sağlar. Kolonilerin sembollerini, taktiklerini ve üretimlerini rastgele atar ve oyunu başlatır.