



T.C.
SAKARYA ÜNİVERSİTESİ

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

Nesne Yönelimli Benzetim – C dili
Rastgele Savaşan Koloniler

B211210031 - Eren KARA

SAKARYA

Mayıs, 2023

Programlama Dillerinin Prensipleri Dersi

Nesne Yönelimli Benzetim – C dili

Rastgele Savaşan Koloniler

Eren Kara

Numara: B211210031 Grup: 1.C

Özet

Bizden C dilinde girilen sayı dizisini boşluklara göre ayırarak her bir sayıya karşılık istenilen şekilde koloni oluşturmamız ve bunu Nesne Yönelimli Benzetim kullanarak gerçekleştirmemiz istendi. Girilen sayılar kolonilerin popülasyonunu temsil ediyordu. Kolonilerin popülasyonları ve yemek stokları vardı. Herhangi biri 0'ın altına düştüğü takdirde ölüyorlardı. Taktik ve Üretim Nesne Yönelimli Benzetim şeklinde tasarlanmış sınıfları ve bu sınıfların savaş() fonksiyonu olmalıydı. Bu savaş() fonksiyonu rastgele bir sayı döndürecek ve diğer kolonilerle savaşta gücünü temsil edecekti. Savaş güçlerine göre popülasyonlarda azalma olacaktı. Kaybedenin yemek stoğunun %10'u kazanan koloniye geçecekti. 1 koloni kalıncaya dek birbirleriyle savaşacaklardı.

© 2023 Sakarya Üniversitesi.

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

Anahtar Kelimeler: C, Nesne Yönelimli Benzetim, Polimorphism benzetimi, Soyut Sınıf Benzetimi, Rastgele Şekilde Savaşan Koloniler

1. GELİŞTİRİLEN YAZILIM

Aşağıda sınıf diye bahsettiğim yapılar derste bize gösterildiği şekilde hazırlanmış nesne yönelimli benzetimli struct'lardır. Her bir struct new_Sınıfismi()-->Constructor ve yokEtSınıfismi()-->Destructor fonksiyonları barındırıyor. Her struct fonksiyonlara içerisinde field olarak barındırdığı fonksiyon göstericisi ile ulaşıyor. Bu fonksiyonlar ise manipüle edeceği struct'ı parametre olarak alıyor.

1.1. Taktik ve Üretim Sınıfları

Bu sınıfları oluşturma için hocamızın videoları ve kaynakça[1]'daki medium makalesinden yararlandım. Makaleden derste anlatılmayan polymorphism nedir onu öğrendim. Ondan sonra Taktik sınıfını kurdum içerisinde savaş() adında 1 tane sanal method barındırıyor. Bu abstract Taktik sınıfının fonksiyonun içerisini doldurabilmek için Taktik'ten kalıtım alacak sınıflara ihtiyaç vardı onları oluşturmaya koyuldum. 2 tane taktik alt sınıfı yaptım: HilalTaktiği ve DaireTaktiği. Bunlar kendi mantıklarına göre savaş() methodunun içerisini dolduruyorlardı.

Üretim sınıfı da Taktik sınıfıyla çok benzer yapıda aralarındaki tek fark birindeki sanal fonksiyonun ismi savaş() diğerinde ise üret(). Avcılık, Balıkçılık, Hayvancılık, Tarım sınıfları bu üret fonksiyonunun içlerini dolduracak Üretim sınıfından kalıtım alan sınıflar oldu. Hepsi kendi mantığıyla üret fonksiyonun içerisini doldurdu ve geriye intger bir değer döndürdü.

1.2. Koloni Sınıfı

Bu sınıf Taktik ve Üretim sınıflarını içerisinde bulunduran onları kullanan bir sınıf olacak. Bilerek kalıtım şeklinde değilde association bağlantısı şeklinde tasarladım. Çünkü bu 2 sınıfı kullanarak işlerini yapacak koloni sınıfı. Kalıtıma gerek olduğunu düşünmedim. Koloni Sınıfı içerisinde, birazdan bahsedeceğim Icerik sınıfın okuduğu sayı dizisine göre belirlenen popülasyon ve yemek değişkenlerini barındırıyor. Ayrıca içerisinde kazanma sayısı, kaybetme sayısı ve rastgele bir sembol field alanları da bulunuyor. İçerisinde savaş sonucunda oluşacak durumları koloniye uygulamak için çeşitli fonksiyonlar barındırıyor.

1.3. Exception Sınıfı

İçerisinde try-catch yapısını gerçekleyen .h dosyasını include etmiş bir şekilde geliyor. Bildiğimiz diğer dillerdeki. Exception sınıflarının C dilindeki hali. Ekstra bir özelliği yok. Eksiği ise kesin vardır.

1.4. Utility Sınıfları

1.4.1 Icerik Sınıfı

İçerisinde, kullanıcıdan okunan stringi, Dizi sınıfından bir nesne (ki bu nesne içerisinde bu stringin sayı dizisine ayrılmış halini ve bu dizinin uzunluğunu tutabiliyor) barındırıyor. Ayrıca IcerikParcala nesnesini tutuyor.

oku() fonksiyonu var bu fonksiyon kullanıcıdan string okuyor ve bu string parçalaması için IcerikParcala'nın parçala() fonksiyonuna veriyor. Bu parçala() fonksiyonu stringi boşluklara göre parçalıyor ve geriye bu parçalanmış sayı dizisini Dizi sınıfına aktarıp geri döndürüyor. Eğerki string'de boşluk ve sayı dışında başka bir şey varsa hata fırlatıyor. Oku() fonksiyonunun içerisinde bu hatayı yakaladım ve kullanıcıdan yeni bir değer istedim. parçala() fonksiyonu içerisinde IcerikKontrol sınıfını barındırıyor ve fonksiyonlarını kullanıyor.

1.4.2 IcerikParcala Sınıfı

İçerisinde Icerik control yapısı bulunuyor ve parçala fonksiyonuna verilen bir stringi integer dizisi şekline ayırıp geri döndürebiliyor.

Parçala() fonksiyonu parameter olarak verilen stringin içerisinde sayıdan başka bir şey varsa hata döndürüyor.

1.4.3 IcerikKontrol Sınıfı

Parametre olarak verilen string değerini sayı olup olmadığını kontrol ediyor.

1.5. Oyun Sınıfları

1.5.1 Olustur Sınıfı

Olustur sınıfı içerisinde Oyun sınıfının basla fonksiyonunda kullanılmak üzere koloni oluşturma fonksiyonu barındırıyor.

Bu fonksiyon girilen sayı dizisindeki sayılar adedince koloni oluşturup geriye koloni dizisi döndürüyor. Bu oluşturma işlemi sırasında yine kendi fonksiyonlarından olan rastgeleTaktik() ve rastgeleUretim() fonksiyonlarını kullanıyor. Bu fonksiyonlar geriye polymorphism şeklinde Taktik ve Uretim sınıflarını döndürüyor. Polymorphsim'i çocuk sınıfın constructor'ını çağırıp (Super*)(çocuk->super) şeklinde dönüştürerek yaptım.

1.5.2 OyunTuru Sınıfı

İçerisinde tur mantığını bulunduran sınıftır. Bu tur mantığını gerçeklerken Savaslar sınıfını da kullanılır. Yani Savaslar sınıfı nesnesine field olarak sahiptir. Ayrıca içerisinde tur içinde gerçekleşen eşleşmeleri tuttuğu eşleşmeA, eşleşmeB sayı dizisine sahip ayrıca Savaslar sınıfının geri döndürdüğü eşleşme ikilisini barındırdığı eşleşmeler field'ına ve son olarak kolonilerin sayısını tutan int değışkene sahip.

tur() fonksiyonu içerisinde savaşSayisi (yasayan koloni sayısının 2'li permütasyonu) kadar dönen ve kolonileri kendi aralarında rastgele bir şekilde savaştıran for döngüsüne sahip. Her bir for iterasyonunda savaşSayisi güncelleniyor çünkü savaşıma sırasında bir koloni ölmüş olabilir. Rastgele eşleşmeleri Savaslar sınıfının eşleşme() fonksiyonu gerçekleştiriyor. Bu fonksiyon geriye 2 tane int değer döndürüyor. Bu for döngüsünde dönen bu değerlere göre bu eşleşen kolonilerin savaş fonksiyonlarını çağırıp kolonileri savaştırmış oluyor. Sonra savaşı sonuçlarının uygulanması için Savaslar fonksiyonundan yardım alıyor. En son olarak da bütün kolonilerin ureme ve yemek tüketme fonksiyonlarını çağırıyor.

1.5.3 Savaslar Sınıfı

İçerisinde tur sonu gerçekleşen olayların ve savaşıacak 2 koloninin rastgele bir şekilde seçilmesini sağlayacak fonksiyonlara sahip. savaşıSonundaYapılacaklar() fonksiyonu Kaybeden koloninin popülasyonunu eksiltiyor. Kaybeden koloninin yemeğini azaltırken, kaybedenden eksilen yemek miktarı kadar kazanan koloniye

yemek veriyor. Kazananın kazanma sayısını ve kaybedenin kaybetme sayısını 1 arttırıyor. eslesme() fonksiyonu içerisinde kolonileri rastgele eşleştiriyor. Bu eşleşmeleri yaparken ölmüş kolonileri seçmiyor. Koloniyi kendisiyle eşleştirmiyor, aynı turda savaşmış kolonileri tekrar savaştırmıyor ve son olarak eslesme sonucunu parametre olarak aldığı bir int dizisine kaydediyor ki tekrar eslesme olacağı zaman aynı turda savaşanları eşleştirmesin. Geriye 2 tane integer değer döndürüyor. Yani savaşacak olan kolonileri.

1.5.4 Oyun Sınıfı

İçerisinde uygulamanın başlatma mantığını barındırıyor. İçerisinde değişken olarak Icerik nesnesi, OyunTuru nesnesi, Koloni dizisi gösteren bir işaretçi, oyun boyunca gerçekleşen toplam tur sayısını barındırıyor.

Bu sınıf basla() fonksiyonuna sahip ki her şey bu fonksiyonla başlayıp bitiyor. basla() fonksiyonunda Icerik sınıfı kullanılarak ilk önce kullanıcıdan kolonilerin popülasyonları alınıyor. Ondan sonra Olustur nesnesi oluşturup bu popülasyonlara göre koloniler oluşturtuluyor. Ondan sonra OyunTuru nesnesi oluşturulup 1 koloni kalana kadar bu nesnenin tur() fonksiyonu çağrılmaya başlanıyor. İşlemler bitince fonksiyon içerisinde heap'ten alınan alanlar geri iade ediliyor.

2. ÇIKTILAR

Zaten tek bir girdi alıyorum. Bu yüzden çok farklı sonuçlarla karşılaşmak mümkün değil. Belki kurduğum algoritmalar beklenmedik şekilde sonuçlar verebilir ancak bir çok denemeye rağmen bir yerlerde sorun göremedim. Onun haricinde kullanıcıdan aldığım ifadeyi yani girdiyi kontrol ettim ve istediğim şekle uygun olmasında kullanıcıyı zorladım.

```
Bir sayi dizisi giriniz: 20 30 40
```

Koloni	Populasyon	YemekStogu	Kazanma	Kaybetme
£	20	400	0	0
¢	30	900	0	0
₤	40	1600	0	0

```
Tur sayisi: 0
```

```
Tur sayisi: 1
```

Koloni	Populasyon	YemekStogu	Kazanma	Kaybetme
£	24	602	2	0
¢	12	705	0	2
₤	45	1431	1	1

```
Bir sayi dizisi giriniz: 20 ke 30
Girilen ifadelerin hepsi sayi olmalı
Bir sayi dizisi giriniz: █
```

```
Bir sayi dizisi giriniz: 20.3 40.6 200 300
Girilen ifadelerin hepsi sayi olmalı
Bir sayi dizisi giriniz: █
```

```
Bir sayi dizisi giriniz: 20 ke
Girilen ifadelerin hepsi sayi olmalı
Bir sayi dizisi giriniz: █
```

3. SONUÇ

Nesneye Dayalı Programlama'nın arkasında yatan mantığı biraz daha iyi kavradım diyebilirim. Kalıtım'ın kullanım şeklini olsun, sanal fonksiyonun çalışma şeklini olsun ve Polymorphism'in çalışma şeklini olsun daha iyi anladım. Örneğin Polymorphism'de base sınıfa çevirmemize rağmen arkada bir yerlerde çocuk sınıf duruyor ve base'e tip dönüştürmesi yaparak bu çocuk sınıfın fonksiyonlarına erişebiliyoruz. Bu baya ilginçti. C'de sınıf olmaması ve her şeyin fonksiyonlarla ve parametreleriyle halledilmesi baya uğraştırıcıydı. NDP kavramının önemini son derece kavratan bir ödevdi. Son olarak da bu kadar şeyi benim yerime yapan complier'a da ayrı bir teşekkürler yani.

Referanslar

- [1] <https://markkevinbaltazar.medium.com/c-language-how-to-implement-polymorphism-ada293bb0377>
- [2] <https://stackoverflow.com/questions/23023458/how-to-read-multiple-integers-from-a-single-line>
- [3] <https://stackoverflow.com/questions/27234795/how-to-cast-a-pointer-to-void-which-is-nested-in-a-struct>
- [4] <https://www.youtube.com/watch?v=bchTIMYSpuE>
- [5] <https://stackoverflow.com/questions/9210528/split-string-with-delimiters-in-c>
- [6] <https://www.youtube.com/watch?v=TQzS6L6zAs0>