



T.C.
SAKARYA ÜNİVERSİTESİ

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

2. Ödev - Yapay Habitat Oluşturma ve Canlı Türlerinin Dinamik Davranışları

B221210042 – ALİ KEREM KOL

SAKARYA

Mayıs, 2024

Programlama Dillerinin Prensipleri Dersi

2. Ödev - Yapay Habitat Oluşturma ve Canlı Türlerinin Dinamik Davranışları

Ali Kerem Kol

^a B221210042 – 1.Öğretim C Grubu

Özet

Ödevde, istenildiği gibi abstract sınıf benzetimi yapılarak ve kalıtım hiyerarşisi kullanılarak “Veri.txt” dosyasından verileri okuyarak değerlere göre “Bitki”, “Bocek”, “Sinek”, “Pire” canlıları oluşturulup işleme tabi tutulmaktadır. Ben canlıları depolamak için iki boyutlu bir dizi kullandım. Ayrıca canlıların türlerini de başka bir matris oluşturarak orada depoladım. Kodum dizi içerisinde gezerek o canlının türüne göre “gorunum()” fonksiyonunu çağırıyor ve ekrana yazdırıyor. Daha sonra kullanıcı bir tuşa bastığında simülasyon başlıyor. Simülasyon “if-else” blokları yardımıyla teker teker tüm olasılıklar kontrol edilip ölen canlı bellekten serbest bırakılıyor. Kodumun çalışma mantığı ilk olarak iç içe for döngüsüyle matrislerde geziyoruz. Daha sonra bulunduğumuz canlıya göre tekrardan matrislerde iç içe for döngüsüyle gezip olasılıkları karşılaştırıyoruz. Bu bizim canlımız bir başka canlı tarafından yok edilene kadar devam ediyor ve sonunda yok olduğunda ise dıştaki iç içe for döngümüzdeki bulunduğumuz konum güncelleniyor ve yeni başka canlılarla karşılaştıracağımız canlımız bizim eski canlımızı yiyen canlı oluyor ve döngü bu şekilde devam ediyor. Son olarak ise kazananı ekrana yazdırıyor ve “delete_habitat” fonksiyonu habitatın içerisinde kalan son canlıyı da bellekten temizleyip habitatı da bellekten temizliyor ve program sonlanıyor. Veri okuyup yerleştirme işlemlerini “main.c” dosyamda diğer; habitat simülasyonu, habitat yazdırması, habitat kazanan yazdırması ve habitat silmesi gibi işlemleri de “habitat.c” dosyamda yaptım. Koduma yorum satırları ekleyerek ve geniş yazarak okunmasını kolay hale getirdim.

© 2017 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Bellek Yönetimi, Kalıtım, Soyut Sınıf, Simülasyon, Veri Yapıları

1. GELİŞTİRİLEN YAZILIM

Ödevime ilk olarak internette araştırmayla başladım, ödev dokümanında verilen iki adet videoyu izledim. İlk olarak “.h” dosyalarını ve “.c” dosyalarını oluşturup kalıtım ve soyut sınıf işlevini yaptım. Soyut fonksiyon olarak da “gorunum()” fonksiyonunu ekledim. Sıra “habitat” sınıfının benzetimini yapmaya geldi, canlıları ve canlı türlerini depoladığım iki boyutlu matrisleri burada oluşturdum. Sonra “main.c” dosyamda “Veri.txt” dosyasından verileri okuyup matrise yerleştirme işlemini gerçekleştirdim. Diğer işlemleri “habitat” sınıfımın içerisinde yaptım (ekrana yazdırma, simülasyon aşaması, habitat silme, kazananı yazdırma vs). Kodumda en çok zorlandığım yer kalıtım hiyerarşisini sağlamak oldu onun dışındaki diğer kısımları daha rahat bir şekilde yaptım. Ayrıca “C” dili orta seviye bir dil olduğu için kodu kontrol etmek, hatamın yerini anlamak oldukça zordu. Özellikle bellek

hatalarında hatanın nerede olduğunun en ufak bir göstergesinin olmaması ve onu tek tek kendim aramış olmam oldukça zamanımı aldı.

2. ÇIKTILAR

Kodum bu girdide:

```
10 2 5 4 7
21 6 1 25 8
11 64 33 41 47
8 1 9 5 23
```

Bu çıktıyı vermektedir:

```
| X| X| X| X| X|
| X| X| X| X| X|
| X| X| X| X| X|
| X| X|B| X| X|
```

Kazanan: B : (3, 2)

3. SONUÇ

Bu kod sayesinde genel olarak soyut sınıf ve kalıtım mantığını çok iyi kavradım. Diğer dillerde de sınıf oluşturduğumuzda ve kalıtım yaptığımızda arka planda nelerin döndüğünü anladım.

Ayrıca ödevi yaparken pratiğimin gelişmesi sayesinde “C” dilindeki pratikliğim ve kod yazma hızım arttı.

Referanslar

- 1) https://www.youtube.com/@MF_A
- 2) https://www.youtube.com/@coder_log