



SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Programlama Dillerinin Prensipleri
Ödev 2

Adı: *Mehmet*

Soyadı: *Bosdancı*

No: G221210045

Grup: 2. Öğretim C Grubu

Öğretmen: *Dr. Öğr. Üyesi HÜSEYİN DEMİRÇİ*

Ödev C dilinde nesne yönelimli benzetimdi. C dilinde class yapıları bulunmadığından class yapılarını struct kullanarak benzetim yaptım. Ödevde Canlı için struct yapısı kullandım burası canlıların ortak özelliklerini tuttuğum bir sınıf. İçerisinde int number ve char* alive alanları bulunuyor. number Veri.txt'den okuduğu sayıları tutuyor. alive ise canlinin hayatta olup olmadığını tutuyor.

Canlı sınıfında dead fonksiyonu var ölen canlılar için return 'X' işlevini yerine getiriyor. delete_Canlı fonksiyonu var bellekten Canlı'yi serbest bırakıyor. gorunum fonksiyonu var Canlı sınıfında gövdesi boş olan soyut fonksiyon mantığıyla çalışıyor hangi canlıının gorunum fonksiyonu çağrılmıysa o canlıının kendine özgü olan gorunum fonksiyonu çalışıyor.

Mesela canlı bitkiyse gorunum_Bitki fonksiyonu çağrılmıyor ,canlı böcekse gorunum_Bocek fonksiyonu çağrılmıyor ,canlı sinekse gorunum_Sinek fonksiyonu çağrılmıyor ,canlı pireyse gorunum_Pire fonksiyonu çağrılmıyor. Çağrılan bu fonksiyonlar geriye hangi canlıya onun baş harfini döndürüyor bitkiyse 'B' gibi.

Bitki struct'ında canlı türüne erişebilmem için Canlı türünden super oluşturduğum böylece Canlı sınıfına erişebiliyorum delete_Bitki fonksiyonunda delete_Canlı fonksiyonunu çağırıyorum daha sonra bitkiyi serbest bırakıyorum

Böcek, sinek ve pire struct'ında da aynı mantıkta super alanı ve delete_Bocek fonksiyonu yer alıyor sinek ve pire oluşturuluyorken önce böcek oluşturuyorum böcekte bana canlı oluşturuyor pire ve sinek için gerekli bağlantıları this->super->super yapısıyla bağlıyorum yani canlı sınıfına bu şekilde erişebiliyorum

Habitat fonksiyonunda Canlı ** matris , satır_sayısı, sutun_sayısı ifadesini tuttum. Bu şekilde canlıları iki boyutlu matriste tuttum. Ayrıca basla_Habitat yemek_habitat yazdır_Habitat delete_Habitat ifadeleri de bulunuyor. basla_Habitat yazdır_Habitat ve yemek_Habitat fonksiyonlarını çağrırmam.

Dosyadan verileri okuyup gelen veriye göre ilgili canlıyı oluşturma işini new_Habitat fonksiyonıyla yapıyorum aynı şekilde bütün struct'larının new fonksiyonu bulunmakta bu fonksiyon genel olarak bellekten yer tahsis edip adresini döndürüyor bize delete_Habitat ise önce hangi canlı varsa o türün delete fonksiyonunu çağrırmam daha sonra matrisdeki tüm canlıların delete_Canlı fonksiyonunu çağrırmam en son ise matris alanını serbest bırakıyorum

Test struct'ında ise main metodunu buluyorum burada habitatattan nesne oluşturulup basla_Habitat ve delete_Habitat fonksiyonları çağrılmıyor

Dosyadan veri okuma işinde zorlandım ve verilere göre canlıları oluşturmada zorlandım önceden direkt cast ediyordum (Canlı)new_Bitki(num) şeklinde ama bu şekilde hatalı oluyordu Bitki bitki = new_Bitki(num) şeklinde önceden oluşturmam gerekiyormuş en çok bu kısımlarda zorlandım.