



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 DEMİRCİ*

Ödev C dilinde nesne yönelimli benzetimdi. C dilinde class yapıları bulunmadığından class yapılarını struct kullanarak benzetim yaptım. Ödevde Canli 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 canlının hayatta olup olmadığını tutuyor.

Canli sınıfında dead fonksiyonu var ölen canlılar için return ‘X’ işlevini yerine getiriyor. delete_Canli fonksiyonu var bellekten Canli’yi serbest bırakıyor. gorunum fonksiyonu var Canli sınıfında gövdesi boş olan soyut fonksiyon mantığıyla çalışıyor hangi canlının gorunum fonksiyonu çağrılıyorsa o canlının kendine özgü olan gorunum fonksiyonu çalışıyor.

Mesela canlı bitkiyse gorunum_Bitki fonksiyonu çağrılıyor ,canlı böcekse gorunum_Bocek fonksiyonu çağrılıyor ,canlı sinekse gorunum_Sinek fonksiyonu çağrılıyor ,canlı pireyse gorunum_Pire fonksiyonu çağrılıyor. Çağrılan bu fonksiyonlar geriye hangi canlıysa onun baş harfini döndürüyor bitkiyse ‘B’ gibi.

Bitki struct’ında canlı türüne erişebilmem için Canli türünden super oluşturdum böylece Canli sınıfına erişebiliyorum delete_Bitki fonksiyonunda delete_Canli 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 Canli ** matris , satir_sayisi, sutun_sayisi ifadesini tuttum. Bu şekilde canlıları iki boyutlu matriste tuttum. Ayrıca basla_Habitat yemek_habitat yazdir_Habitat delete_Habitat ifadeleri de bulunuyor. basla_Habitat yazdir_Habitat ve yemek_Habitat fonksiyonlarını çağırıyor.

Dosyadan verileri okuyup gelen veriye göre ilgili canlıyı oluşturma işini new_Habitat fonksiyonuyla yapıyorum aynı şekilde bütün struct’larımın 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 çağırıyor daha sonra matrisdeki tüm canlıların delete_Canli fonksiyonunu çağırıyor en son ise matris alanını serbest bırakıyor

Test struct’ında ise main metodu bulunuyor burada habitattan nesne oluşturulup basla_Habitat ve delete_Habitat fonksiyonları çağrılıyor

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