

1A - 1A

## Veri Yapıları 1. Ödev Raporu

Öncelikle ödev dokümanında da şeklen tasvir edildiği gibi kova nesnelerini tutan pointer diziye ihtiyaç duyulmaktadır. Bu nesnelerin içerisindeyse ayrıca bir pointer bulunacak ve bu pointerda kovaların içine atanması muhtemel topların adreslerini tutacaktır. Kovaları tutacak pointer dizi, kova nesneleri ve kova nesnelerinde bulunacak olan pointerin tutacağı top nesnesi heap bellek bölgesinde oluşturulacaktır. Programın sınıf tasarımıdaysa kova sınıfı, top sınıfı ve kovakontrol sınıfı bulunacaktır.

Tasarımımızın uygulamasına kova sınıfını oluşturarak başladık. Bu sınıfın içerisinde private olarak tanımlanan alt alanlar oluşturduk. Bu alt alanlar; oluşturulması muhtemel topları tutacak olan topTucucu pointer, kovalarımızın renk bilgisinin bulunacağı renk değişkeni ve kovaların şekillerinin oluşmasını sağlayacağımız 2 boyutlu kova dizisidir. Ardından kova sınıfımızın kurucu metodunda şekli oluşturacak olan dizinin elemanlarına gerekli atamaları yaptık ve kovanın renginin belirlenmesi işlemini rand fonksiyonu yardımıyla tamamladık. Sonrasında kova sınıfının private değişkenlerine dışarıdan erişebilmek amacıyla get ve set metotlarını yazdık. Kovayı bastırabilmek için kovaBastir fonksiyonunu da yazıp kova sınıfını bitirdik.

Kova sınıfının bitmesinden sonra top sınıfını oluşturmaya başladık. Topumuzu temsil edecek olan harfi, oluşturduğumuz harfler dizisinde bulunan 26 karakterden birinin rastgele alınmasını sağlamak için rand fonksiyonunu kullanarak aldık. Harf ataması da yapılan topa dışarıdan erişim sağlayabilmek için get metodu yazdık ve top sınıfını bitirmiş olduk.

Kova ve top sınıflarının bitmesinin ardından kova kontrol sınıfını oluşturmaya başladık. Sınıfın içerisinde private olarak tanımlanan alt alanlar oluşturduk. Bu alt alanlar; kullanıcıdan alınacak olan kovaSayisi ve kova nesnelerini tutacak olan kova pointerıdır. Ardından kova kontrol sınıfının kurucu metodunda kova sayısını aldık ve kova sayısını heap bellek bölgesinde yeterli miktarda yer ayrılabilmesini sağlamak amacıyla kova nesnelerini tutacak olan pointer diziyi tanımlarken kullandık.

Sonrasında oluşturulan bu pointer diziyi, private alanı içerisindeki kovalar pointerına atadık. Ardından ödev dökümanında belirtilen işlemleri yapabilmek amacıyla kova kontrol sınıfı içerisinde metotlarımızı yazmaya başladık.

Öncelikle ilk işlemimizi yapabilmek için `değerKontrol` adında kovaların boş olup olmadığını kontrol eden bir fonksiyon yazdık. Bu fonksiyondan dönen değeri `topOluştur` fonksiyonumuzda eğer top varsa yeniden oluşmasını engellemek için kullandık. Top yoksa `TopOluştur` fonksiyonu içerisindeki for yapısı kovaların kadar dönerken top nesneleri oluşturacak ve oluşan bu nesneleri kova sınıfının içerisinde bulunan `topTutucu` pointera atayacak şekilde kodumuzu yazdık. Ardından oluşturulan bu topları kovalarımızın içerisinde koymak için `toplarıYerleştir` adında bir fonksiyon yazdık. Fonksiyon gövdesinde yine kovaların kadar dönen bir for yapısı yardımıyla kovaların ilgili elemanlarına toplarımızı yerleştirdik.

Ardından ikinci işlem için kullanılacak olan `toplarıYoket` fonksiyonunun gövdesini yazmaya başladık. Bu fonksiyonu, gövdesinde bulunan for yapısı yardımıyla kova nesnelerinde dolaşacak ve topların bulunduğu bölgeleri null ataması ve boş olduğunu göstermek amacıyla kullandığımız '-' karakteriyle dolduracak şekilde yazdık.

Sonrasında top değiştir fonksiyonunu yazdık. Fonksiyonun gövdesinde öncelikle hangi iki topun değiştirileceğini kullanıcıdan aldık. Swap işlemini uygulayacağımız için veri kaybını engellemek amacıyla geçici bir top türünde değişken tanımladık. İlk topu oluşturduğumuz geçici değişkene, ikinci topu ilk topun yerine sonrasındaysa geçici değişkende tuttuğumuz ilk topu ikinci topun eski konumuna yerleştirerek işlemi tamamladık. Tüm bu işlemleri `kovaDeğiştir` fonksiyonu için de yazdık.

Daha sonra `toplarıTersCevir` fonksiyonunu yazdık. Top oluştur fonksiyonunda kullandığımız gibi burada da eğer toplar yoksa fonksiyon gövdesinin çalışmamasını sağlayacak gerekli if yapısını yazdık. Eğer toplar varsa bu topları geçici olarak tutacak olan bir pointer diziyi ters olarak yerleştirme işlemini yapacak olan bir for yapısı yazdık. Sonrasında geçici pointer dizide ters olarak bulunan topları yeniden kovalar içerisindeki `topTutucu`ya atadık. Böylelikle toplar ters çevrilmiş oldu. Aynı işlemleri `kovalarıTersCevir` fonksiyonunda da kullandık.

Son olarak heap bellek bölgesinde ayırdığımız alanları bilgisayara geri teslim etmek amacıyla delete komutlarının bulunduğu fonksiyonumuzu yazdık.

Tüm sınıfların oluşturulması bittikten sonra kullanıcının sistemi kullanabilmesi için test programında tanımladığımız main fonksiyonunda, do while yapısı içerisinde switch case yapısı kullanarak oluşturduğumuz menüyü yazdık.

Ödevi oluştururken en zorlandığımız kısım heap bellek bölgesinden yer ayırmak oldu. Yer ayırma işlemini kurucu fonksiyonda yaparak bu sorunu çözmeyi başardık.