**YZM 2124**

Veri Yapıları

**ÖDEV\_2 # RAPORU**

Öğrenci Adı: Hakan Bayazıt Habeş

**1.Giriş**

Ödevin amacı bağlantılı liste kullanılarak bir kuyruk veri yapısı tasarlanacaktır. Daha sonra, tasarlanan kuyruk yapısı kullanılarak bir yığın veri yapısı tasarlanacaktır. Son olarak, özyinelemeli programlama kullanılarak yığının içerisinde yer alan veriler büyükten küçüğe doğru sıralanacaktır.

\* Tasarlayacağımız kuyruk veri yapısına ilişkin aşağıda verilen tanımları (Metot başlıklarını) kullanmaktayız.

typedef int KuyrukVeriTipi;

struct Dugum

{

KuyrukVeriTipi veri; 🡪 düğümün sayısal değeri

Dugum\* sonraki; 🡪 sonraki düğüme işaretçi

};

struct Kuyruk

{

Dugum\* bas; 🡪kuyruğun başındaki elemanın adresini tutar

Dugum\* son; 🡪kuyruğun sonundaki elemanın adresini tutar

void olustur(); 🡪başlangıç için gereken tanımları içerir

void kapat(); 🡪program bitiminde bellek iadesi yapar

void ekle(int); 🡪kuyruğa yeni veri ekler

int cikar(); 🡪kuyruktan sıradaki elemanı çıkarır

bool bosMu(); 🡪kuyruk boş mu kontrol eder

int size(); 🡪kuyruğun boyutunu tutar

};

\*Oluştur() ve kapat() metotlarının oluşturulması \*bosMu() ve size() metotlarının oluşturulması

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

\* ekle(int) ve cikar() metotlarının oluşturulması

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

struct Yigin {

Kuyruk k1, k2; 🡪 yığın için kullanılacak kuyruk değişkenleri

Dugum\* bas; 🡪Yığının bas adresini tutar

void olustur(); 🡪başlangıç için gereken tanımları içerir

void kapat(); 🡪program bitiminde bellek iadesi yapar

void ekle(int); 🡪kuyruğa yeni veri ekler

int cikar(); 🡪kuyruktan sıradaki elemanı çıkarır

bool bosMu(); 🡪kuyruk boş mu kontrol eder

int tepe(); 🡪yığının tepesindeki elemanı okur

void yazdir(); 🡪yığın elemanlarını sırayla ekrana yazar };

\* olustur() ve kapat() metotlarının oluşturulması \* ekle(int) metodunun oluşturulması

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

\* cikar() , bosMu() , yazdir() ve tepe() metotlarının oluşturulması

metin içeren bir resim

Açıklama otomatik olarak oluşturuldumetin içeren bir resim

Açıklama otomatik olarak oluşturuldu

\*yiginCikar(Yigin\* s) metodunun oluşturulması \*siraliEkle(Yigin\* s,int x) metodunun oluşturulması

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

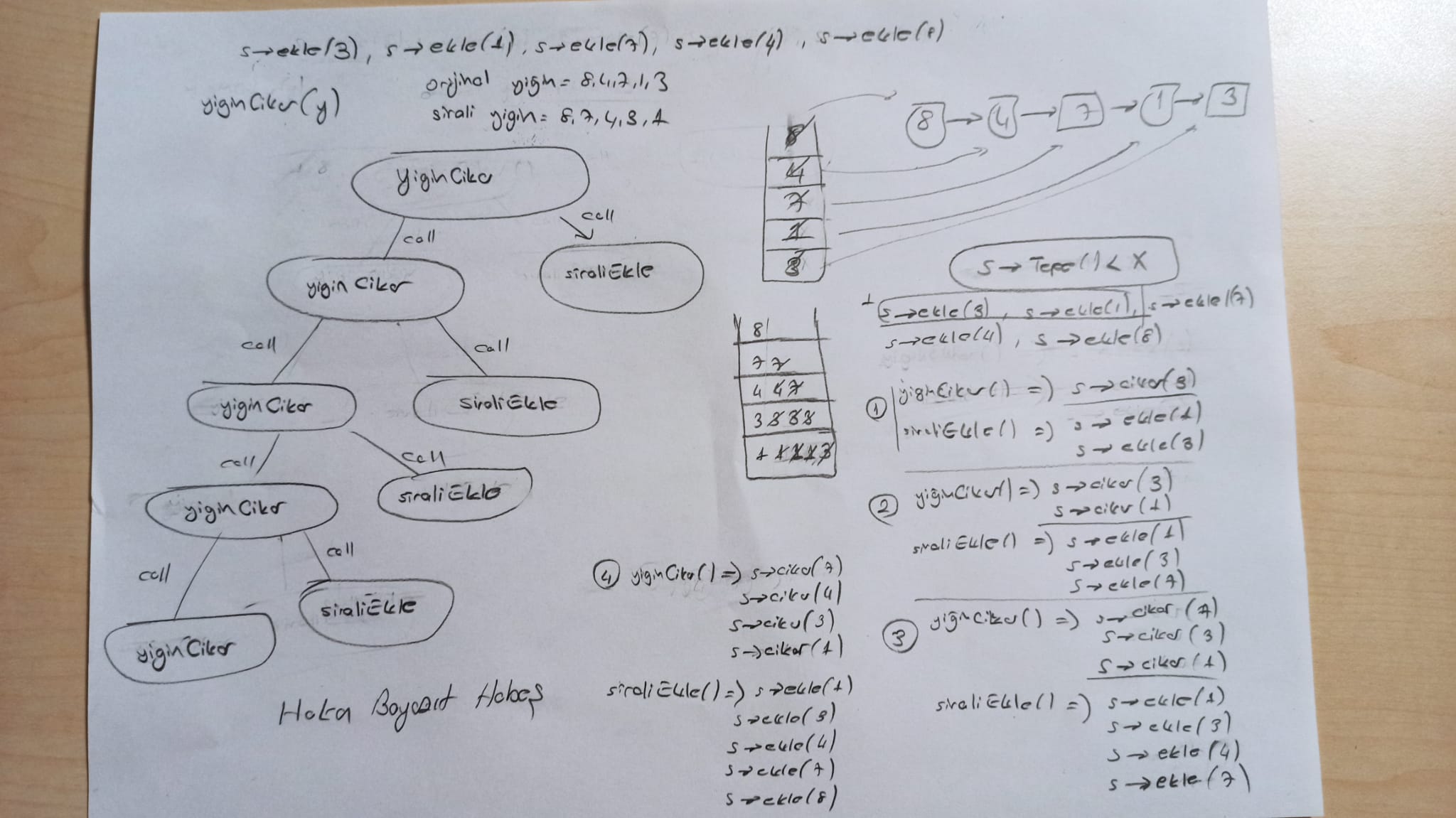
**2. Geliştirme ve Çalıştırma Ortamları**

Ödev gerçekleştirilirken Windows makine üzerinde ve Visual studio üzerinde oluştur, ekle, güncelle, sil, yazdır, boşalt metotları kullanılarak geliştirilmiştir.

**3. Veri Yapıları ve Değişkenler**

Ödevin gerçekleştirilmesinde aşamasında linkedlist ,kuyruk ve yığın veri yapıları kullanılmıştır. Kuyruk ekleme işleminde **First-In-FirstOut(FIFO)** prensibi vardır. Yani ilk giren eleman, ilk çıkar. Örneğin sinema bileti almak için sıraya girmiş kişileri düşünebiliriz. Yığın veri yapısında ise **Last-In-First-Out (LIFO)** prensibi vardır. Yani son giren eleman, ilk çıkar. Örneğin üst üste dizilen kitapları, yalnızca en üsttekine erişecek şekilde düşünebiliriz. Bu yapılar bizim uygulamamız için en uygun veri yapılarıdır.

**4. Program Akışı**



**Programın En son Ekran Görüntüsü**

**metin içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**5. Sonuç**

Bu ödevden öğrendim kazanımlardan birisi yapacağımız uygulamalarda hangi veri yapısını kullanmam gerektiğine dair bir fikir verdi. Kuyruk ve yığının dezavantajlarını detaylı bir şekilde öğrenme imkânı buldum. Bağlantılı liste kullanarak kuyruk veri yapısının ne şekilde tasarlanması gerektiğine dair fikir sahibi oldum. Burada elde ettiğim kazanımları gerçekleştireceğim projelerde daha detaylı ele alacağım.