



BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
BM208 VERİ YAPILARI VE ALGORİTMALAR DERSİ
UYGULAMA FÖYÜ



7. Deney Föyü: Kuyruk Yapısı

Adı Soyadı:

Öğrenci No:

26.03.2019

1. Amaç:

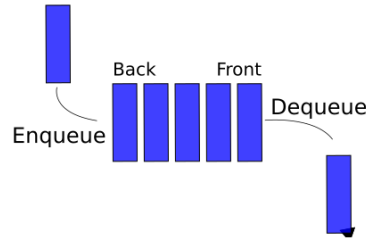
Kuyruğun tanımı, kuyruğa ekleme, çıkarma ve arama işlemleri

2.Uygulama Dersinden Önce Yapılması Gerekenler

Kuyruk yapısının önerilen kitaplardan incelenmesi gerekmektedir. Kuyruğa eleman ekleme ve çıkartma işlemlerinin nasıl yapıldığının araştırılması gerekmektedir.

3.Uygulamaya Hazırlık:

Tek yönlü bağlı liste kullanarak kuyruk yapısını bulabilirsiniz. Kuyruk yapısı dediğimizde aklımıza FIFO (first in first out) gelmesi gerekiyor. İlk giren ilk çıkar yapısı ile kuyruğa ilk giren veri kuyruktan ilk çıkar. En son giren veriyi elde etmek için önündeki tüm verilerin çıkması gerekir. Bu yapı ATM örneğine benzemektedir. ATM'den para çekmek için sıraya girdiğimizde ilk sıradaki işini bitirir ve sıradan çıkar. En sondaki kişi ise önündeki insanların işini bitirmesini bekler ve işini en son o bitir.



```
#include <iostream>
#include <cstdlib>
#include <string.h>
using namespace std;
void secim(int secilen);
//Düğümde ad soyad ve yaş değerlerini tutuyoruz.
struct dugum{
    char isim[20];
    char soyIsim[20];
    int yas;
    struct dugum *next;
};
//Bas: Kuyruğun ilk elemanı
//kuyruk: kuyruğun son elemanı
//temp: fonksiyonlarda kullanabilmemiz için geçici değişken
struct dugum* bas = NULL;
struct dugum* kuyruk = NULL;
struct dugum* temp;
//İsim, soyisim ve yaş için global değişkenler
char isim[20], soyIsim[20];
int yas;
//Bu fonksiyon düğüm oluşturur ve return eder
struct dugum* kisiOlustur(char isim[], char soyIsim[], int yas)
{
    struct dugum* yenikisi = (struct
dugum*)malloc(sizeof(struct dugum));
    strcpy(yenikisi->isim, isim);
    strcpy(yenikisi->soyIsim, soyIsim);
    yenikisi->yas = yas;
    yenikisi->next = NULL;
    return yenikisi;
};
//Kuyruğa eleman ekleme (enqueue)
void elemanEkle(char isim[], char soyIsim[], int yas)
{
    struct dugum* kisi = kisiOlustur(isim, soyIsim, yas);
    //Eğer ilk eleman ve son eleman boş ise, yeni gelen eleman
    ilk elemandır
```

```
if(bas == NULL && kuyruk == NULL)
{
    bas = kisi;
    kuyruk = kisi;
} else {
    //Sonuncunun arkasına eleman eklenir
    kuyruk->next = kisi;
    //Sonuncu eleman yeni eklenen eleman
    olacak şekilde güncellenir.
    kuyruk = kisi;
}
//Kuyruktan eleman çıkaran fonksiyon
void elemanSil()
{
    temp = bas;
    //Eğer kuyruğun ilk elemanı yoksa kuyruk zaten
    boştur.
    if(bas == NULL)
    {
        cout<<"\nKuyruk bos, lutfen eleman
ekleyiniz";
        return;
    }
    //Eğer kuyruğun en arkasındaki kişi aynı zamanda
    en önündeki kişi ise kuyrukta 1 kişi var demektir
    //Bu yüzden o kişi de kuyruktan çıktığında
    kuyruğun hiçbir elemanı kalmaz.
    if(bas == kuyruk)
    {
        bas = NULL;
        kuyruk = NULL;
        return;
    }
    //Eğer yukarıdaki iki ihtimalde sağlanmazsa
    aşağıdaki kısım çalışır.
    //Mantık kuyruğun ikinci sırasındaki kişi
    artık birinci (bas) sırasına gelmiş olur
    bas = temp->next;
    //Free komutu ile düğümü yok ediyoruz
    free(temp);
}
```

```

//Sırada kimin olduğunu gösteren fonksiyon
struct dugum* siradaki()
{
    if(bas == NULL)
    {
        cout<<"\nKuyrukta eleman yok...";
        return 0;
    }

    return bas;
}
//Kuyruğu yazdıran fonksiyon
void kuyrukYaz()
{
    int i = 1;
    if(bas == NULL)
    {
        return;
    }
    temp = bas;
    while(temp->next != NULL)
    {
        cout<<i<<" . Konum " <<temp->isim<<" "<< temp-
>soyIsim<<" " <<temp->yas <<endl;
        temp = temp->next;
        i++;
    }
    cout<<i<<" . Konum " <<temp->isim<<" "<< temp-
>soyIsim<<" " <<temp->yas <<endl;
}
//Menu secimi
void secim(int secilen)
{
    switch(secilen)
    {
        case 1:
            cout<<"\n Isim giriniz ... ";
            cin>>isim;
            cout<<"\n Soyisim giriniz ... ";
            cin>>soyIsim;
            cout<<"\n Yas giriniz ... ";
            cin>>yas;
            elemanEkle(isim, soyIsim, yas);
            kuyrukYaz();
            break;
        case 2:
            elemanSil();
            kuyrukYaz();
            break;
        case 3:
            temp = siradaki();
            cout<<"\n ***** \n";
            if(temp != NULL)
            {
                cout<<temp->isim<<" "<< temp->soyIsim<<" "
<<temp->yas <<endl;
            }

            break;
    }
}
//Main fonksiyonu, yalnızca menu fonksiyonunu çağırıyoruz.
int main()
{
    int secilen;
    while( 1 == 1 )
    {
        cout<<"\n 1- Ekle ... ";
        cout<<"\n 2- Cikart ... ";
        cout<<"\n 3- Sıradaki eleman hangisi? ";
        cout<<"\nLutfen seciminizi yapiniz ";
        cin>>secilen;
        secim(secilen);
    }
    return 0;
}

```

Uygulama Sorusu : Kuyruktaki elemanlar arasından yaşı çift olan 2. Kişiyi bulan ve bilgilerini ekrana yazan fonksiyonu yazınız

CEVAP :

Değerlendiren Öğretim Elemanı:		Çalışma Durumu:	0	1	2	3	4	5	Not:
-----------------------------------	--	--------------------	---	---	---	---	---	---	------