İSİM : NATHANAELLE BOPTI NGAH BONG

ÖGRENCİ NUMERASI: 24080410150

BİLGİSAYAR MÜHENDİSLİĞİ

```
using System;

public class CustomerNode
{
    public int Code;
    public string FullName;
    public CustomerNode NextCustomer;

    public CustomerNode(int code, string fullName)
    {
        Code = code;
        FullName = fullName;
        NextCustomer = null;
    }
}

public class MultiPriorityQueue
{
    private CustomerNode[] frontNodes;
    private CustomerNode[] rearNodes;
    private int priorityLevels;

    public MultiPriorityQueue(int totalPriorities)
    {
        priorityLevels = totalPriorities;
        frontNodes = new CustomerNode[priorityLevels];
        rearNodes = new CustomerNode[priorityLevels];
    }

    public void AddCustomer(int code, string fullName, int priorityGroup)
    {
        int index = priorityGroup - 1;
        CustomerNode newCustomer = new CustomerNode(code, fullName);

        if (frontNodes[index] == null)
```

```
        {
            frontNodes[index] = newCustomer;
            rearNodes[index] = newCustomer;
        }
        else
        {
            rearNodes[index].NextCustomer = newCustomer;
            rearNodes[index] = newCustomer;
        }

        Console.WriteLine($"[{priorityGroup}] - {fullName} (#{code}) kuyruğa
eklendi.");
    }

    public void ServeCustomer()
    {
        for (int i = 0; i < priorityLevels; i++)
        {
            if (frontNodes[i] != null)
            {
                CustomerNode served = frontNodes[i];
                frontNodes[i] = frontNodes[i].NextCustomer;

                if (frontNodes[i] == null)
                    rearNodes[i] = null;

                Console.WriteLine($"Öncelik {i + 1} -> {served.FullName}
(#{served.Code}) kuyruktan çıkarıldı.");
                return;
            }
        }

        Console.WriteLine("Şu anda bekleyen müşteri bulunmamaktadır.");
    }
}

public class CustomerRecord
{
    public int Code;
    public string FullName;
}

public class ArrayBasedPriorityQueue
{
    private int[] frontIndices;
    private int[] rearIndices;
    private int[] totalItems;
    private int totalPriorities;
    private int maxCapacity;
    private CustomerRecord[,] records;

    public ArrayBasedPriorityQueue(int priorities, int maxSize)
    {
        totalPriorities = priorities;
        maxCapacity = maxSize;
        frontIndices = new int[totalPriorities];
        rearIndices = new int[totalPriorities];
        totalItems = new int[totalPriorities];
```

```
        records = new CustomerRecord[totalPriorities, maxCapacity];
    }

    public void AddCustomer(int code, string fullName, int priorityGroup)
    {
        int index = priorityGroup - 1;

        if (totalItems[index] == maxCapacity)
        {
            Console.WriteLine($"Öncelik {priorityGroup} kuyruğu dolu!");
            return;
        }

        records[index, rearIndices[index]] = new CustomerRecord { Code = code,
FullName = fullName };
        rearIndices[index]++;

        if (rearIndices[index] == maxCapacity)
            rearIndices[index] = 0;

        totalItems[index]++;
        Console.WriteLine($"[{priorityGroup}] - {fullName} (#{code}) başarıyla
eklendi.");
    }

    public void ServeCustomer()
    {
        for (int i = 0; i < totalPriorities; i++)
        {
            if (totalItems[i] != 0)
            {
                int currentFront = frontIndices[i];
                string fullName = records[i, currentFront].FullName;

                frontIndices[i]++;
                if (frontIndices[i] == maxCapacity)
                    frontIndices[i] = 0;

                records[i, currentFront] = null;
                totalItems[i]--;

                Console.WriteLine($"{fullName} adlı müşteri kuyruktan çıkarıldı.");
                return;
            }
        }

        Console.WriteLine("Tüm kuyruklar boş, işlem yapılacak müşteri yok.");
    }
}

class DemoApp
{
    static void Main()
    {
        Console.WriteLine("Bağlantılı liste - öncelik sırasına göre \n");

        MultiPriorityQueue listQueue = new MultiPriorityQueue(3);
        listQueue.AddCustomer(1035, "Telor", 3);
```

```
        listQueue.AddCustomer(1039, "Swift", 2);
        listQueue.AddCustomer(1049, "Raymon", 3);
        listQueue.AddCustomer(1000, "Nathy", 1);

        Console.WriteLine();
        listQueue.ServeCustomer();
        listQueue.ServeCustomer();
        listQueue.ServeCustomer();
        listQueue.ServeCustomer();
        listQueue.ServeCustomer();

        Console.WriteLine("\n-----------------------------------------\n");

        Console.WriteLine(" Dizi tabanlı öncelik sırası\n");

        ArrayBasedPriorityQueue arrayQueue = new ArrayBasedPriorityQueue(3, 10);
        arrayQueue.AddCustomer(7601, "Nathy", 2);
        arrayQueue.AddCustomer(7652, "Clara", 1);
        arrayQueue.AddCustomer(7603, "Cebir", 3);
        arrayQueue.AddCustomer(7604, "Merlin", 1);

        Console.WriteLine();
        arrayQueue.ServeCustomer();
        arrayQueue.ServeCustomer();
        arrayQueue.ServeCustomer();
        arrayQueue.ServeCustomer();
        arrayQueue.ServeCustomer();
    }
}
```

```
Microsoft Visual Studio Debu

Baglantili liste - öncelik sirasina göre

[3] - Telor (#1035) kuyruga eklendi.
[2] - Swift (#1039) kuyruga eklendi.
[3] - Raymon (#1049) kuyruga eklendi.
[1] - Nathy (#1000) kuyruga eklendi.

Öncelik 1 -> Nathy (#1000) kuyruktan çikarildi.
Öncelik 2 -> Swift (#1039) kuyruktan çikarildi.
Öncelik 3 -> Telor (#1035) kuyruktan çikarildi.
Öncelik 3 -> Raymon (#1049) kuyruktan çikarildi.
Su anda bekleyen müsteri bulunmamaktadir.

-----------------------------------------

 Dizi tabanli öncelik sirasi

[2] - Nathy (#7601) basariyla eklendi.
[1] - Clara (#7652) basariyla eklendi.
[3] - Cebir (#7603) basariyla eklendi.
[1] - Merlin (#7604) basariyla eklendi.

Clara adli müsteri kuyruktan çikarildi.
Merlin adli müsteri kuyruktan çikarildi.
Nathy adli müsteri kuyruktan çikarildi.
Cebir adli müsteri kuyruktan çikarildi.
Tüm kuyruklar bos, islem yapilacak müsteri yok.

C:\Users\Nathy\source\repos\Vedat_odevi\bin\Debug\net8.0\Vedat_odevi.e
To automatically close the console when debugging stops, enable Tools-
```