**PROJE 2 : LİSTE, YIĞIT, KUYRUK ve ÖNCELİKLİ KUYRUK VERİ YAPILARI**

**Proje tek kişiliktir. Her bir öğrenci kendi başına hazırlayacaktır. İstenen programlar ilgili öğrenci tarafından yazılıp tamamlandıktan sonra rapor, kodlar arka arkaya getirilip her maddede istenen bilgiler de yazılarak tamamlanacaktır. Raporda, her bir soru için soru numarası, ilgili kaynak kod, istenen diğer bilgiler ve ekran görüntüsü yer almalıdır. Programlar ve Rapor, belirtilecek tarihe kadar EGEDERS’te Proje 2 Yükleme bağlantısından yüklenmelidir. Doğru çalışmasa bile, kendi başınıza yapmanıza teşvik açısından, fazla puan kırılmayacaktır. Kopya ödev ise istenmemektedir. Gerekirse bazı öğrenciler Teams üzerinden kontrole çağrılabilir. C# veya Java tercih edilmelidir.**

1. **a)** **Bileşik Veri Yapısı Oluşturma ve Eleman Ekleme Ön Çalışması ve Metodu:** Dizilerdeki (MahalleAdı, TeslimatSayısı) bilgilerini, Şekil 1’deki Bileşik Veri Yapısına, gösterildiği biçimde yerleştiren (for döngüsünden yararlanarak) **etkin C#** veya Java programını **kurşunkalemle A4 boyutunda bir kağıda yazarak (20 puan)** raporun başına ekleyiniz. Dinamik dizinin (**ArrayList**) her bir elemanının Mahalle sınıfı yapısında olmasına; MahalleAdı bilgisi ve (Teslimat sınıfı nesneleri içeren) **Generic List’ten** oluşmasına dikkatediniz. Öncelikle **Teslimat sınıfını** (**Yemek Adı, Adet** sahalarını içeren) oluşturunuz. Sonra **ArrayList içindeki Generic List’lerin her birini ilgili mahalledeki Teslimat sayısı kadar Teslimat nesnesi ile doldurunuz. Yemek bilgilerini, bir yemek listesi oluşturup oradan rastgele (random) seçebilirsiniz. Bu proje için her bir teslimatın sadece bir yemek türünden (kaç adet olacağı da dahil) oluşabileceğini varsayın.**

**a maddesinde yazacağınız metot, herhangi bir dizi boyutu için çözüm üretilebilmelidir. Örnek olarak MahalleAdı ve TeslimatSayısı dizileri 50 elemanlı ise onun için de çözüm üreten bir metot yazılmalıdır. MahalleAdı ve TeslimatSayısı** dizilerinin aynı uzunlukta verildiği varsayılabilir.

**b)** Daha sonra kodlayarak çalıştırınız. Aşağıdaki dizileri kullanarak oluşturduğunuz Bileşik veri yapısındaki elemanları yazdırınız. **(20 puan)**

MahalleAdı = { Özkanlar, Evka 3, Atatürk, Erzene, Kazımdirik, Mevlana, Doğanlar, Ergene }

TeslimatSayısı = { 5, 2, 7, 2, 7, 3, 0, 1 }

Teslimat

…

Generic Lists

Pilav, 1

Pizza, 5

Türlü, 2

…

…

…

…

Simit, 7

Börek, 2

…

…

Kızartma, 15

ArrayList  
Moto\_Kurye

1

**…**

Evka 3,

0

2

3

Mahalle

**Şekil 1:** Liste’lerden oluşan ArrayList (Bileşik Veri Yapısı)

**c) Dinamik Dizinizde (ArrayList) kaç tane eleman (Liste) olduğunu buldurup yazdırınız (Şekil 1’deki örneğin tamamı için 8). Bileşik Veri Yapısındaki toplam teslimat sayısını buldurup ekrana yazdıran kodu yazınız (Şekil 1 için 27). Rapora toplam teslimat sayısı sonucunuzu da ekleyiniz. (5)**

**2)** **YIĞIT ve KUYRUK** **(10 puan)**

a) Ders kitabındaki Chapter 4’te LISTING 4.1’deki **yığıt** (sayfa 120-121) programını inceleyiniz, ilgili bölümleri okuyunuz. Mahalle (sınıfı nesnelerinden oluşan) Yığıtı oluşturacak şekilde kodu güncelleyiniz veya kendiniz yazınız. Soru 1’deki tüm mahalleleri, oluşturduğunuz Yığıta ekleyiniz (bilgileri dizilerden de, oluşturduğunuz ArrayList’ten de çekebilirsiniz, size kalmış). Yığıttaki tüm elemanları çıkartıp ekrana mahalle bilgilerini (Mahalle Adı ve listesindeki tüm teslimat bilgileri) ekrana yazdırınız. (5)

b) Ders kitabındaki Chapter 4’te LISTING 4.4’teki **kuyruk** (138-140) programını inceleyiniz, ilgili bölümleri okuyunuz. 2a.’daki işlemi **Kuyruk** (sınıfı) Veri Yapısı için tekrarlayınız. (5)

**3) ÖNCELİKLİ KUYRUK (15 puan)**

**a)** C# veya Java’da **azalan** sırada ve O(1) ekleme zamanlı (ama en öncelikli elemanı silmenin daha yavaş olduğu) bir **Öncelikli Kuyruk (sınıfı)** tasarlayarak yazınız (Liste sıralı tutulmayacak, eleman sona eklenecek, eleman silme metodu ise **en fazla teslimat yapılacak** olan mahalleyi arayarak onu silecek). **Mahalle sınıfı** tipindeki elemanları tutmak için **List** hazır veri yapısı kullanmalısınız. **ÖncelikliKuyruk** Sınıfının gerekli tüm metotlarını (ekle, sil, bosMu, yapılandırıcı) ve içeriklerini yazmalısınız. Soru 1’deki veriler üzerinde test ediniz, mahalleleri teslimat sayısına göre öncelikli şekilde sırayla silerek (büyükten küçüğe) yazdırınız. (10 )

**İpucu :** Yeni gelen elemanı, altyapıda kullandığınız listenin sonuna ekleyebilirsiniz ve en büyük değere sahip yani en öncelikli elemanı **arayarak** silebilirsiniz.

**Not:** Elemanları Dizi‘de tutan yarı puan alır.

b) Öncelikli Kuyruğun gerçekleştiriminde altyapıda mahallelerin tutulmasında List yerine dizi kullanılırsa ne gibi etkinlik sorunları ortaya çıkar? Kısaca açıklayınız. (5)

**4) (20 puan)**

**a) Her bir elemanı tamsayı olan bir kuyruk tasarlayarak yazınız.** Bir marketteki tek kasada bekleyen müşterilerin sepetlerinde sıra ile **6, 7, 2, 1, 12, 5, 3, 7, 4, 2** adet ürün olsun. Kasiyerin her bir ürünü okutmasının **3 saniye** sürdüğünü varsaydığınızda **Kuyruk (FIFO yapısındaki)**’taki her bir müşterinin **işlem tamamlanma sürelerini** ve bu kasa için müşterilerin ortalama işlem tamamlanma süresini bulunuz (ödeme gibi işlemlerin süresini ihmal edebilirsiniz). (5)

**b)** 3. Sorudaki **ÖncelikliKuyruk** sınıfı ve metotlarını tamsayılar için artan sırada işleyecek şekilde güncelleyiniz. 3. sorudaki sınıfınızı kaybetmemek için yedekleyiniz. Bu **ÖncelikliKuyruk** sınıfını kullanarak veri yapısındaki **her bir müşterinin işlem tamamlanma sürelerini** ve bu kasa için **ortalama işlem tamamlanma** süresini bulunuz, sonuçları rapora ekleyiniz. (10)

**c)** Kasalarda Queue ve PQ ve yapıları kullanımının sonuçlarını kısaca karşılaştırınız. Öncelikli Kuyruk (PQ) düzeni, ortalama işlem tamamlanma süresi açısından daha verimli olmaktadır. Ancak dezavantajları nelerdir? Hangi durumlarda kullanılamaz. Kısaca açıklayınız. (5)

**5) Özdeğerlendirme tablosunu doldurarak raporun sonuna ekleyiniz (10 puan):**

**Özdeğerlendirme Tablosu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Proje 2 Maddeleri** | **Puan** | **Tahmini Not** | **Açıklama** |
| **1 a) A4 Ön çalışma** | **20** |  |  |
| **1 b) Kaynak kod, ekran görüntüsü, veri yapısının elemanlarının listelenmesi** | **20** |  |  |
| **1 c) Kaynak kodlar, Liste sayısı, veri yapısındaki toplam teslimat sayısı** | **5** |  |  |
| **2 a) Yığıt kaynak kod ve ekran görüntüleri** | **5** |  |  |
| **2 b) Kuyruk kaynak kod ve ekran görüntüleri** | **5** |  |  |
| **3 a) Öncelikli Kuyruk kod ve ekran görüntüleri** | **10** |  |  |
| **3 b) ArrayList ve Dizi altyapılarının karşılaştırılması** | **5** |  |  |
| **4) Kod, sonuçlar tablosu, ekran görüntüleri, karşılaştırma ve soruların cevapları.** | **20** |  |  |
| **5) Özdeğerlendirme Tablosu** | **10** |  |  |
| **Toplam** | **100** |  |  |

**Açıklama kısmında yapıldı, yapılmadı bilgisi ve hangi maddelerin nasıl yapıldığı (ve nelerin yapılmadığı / yapılamadığı) yazılmalıdır. Tahmini not kısmına da ilgili maddeden kaç almayı beklediğinizi yazmalısınız.**

**Not: Konsol uygulaması olarak yapan öğrenciler Console ekranlarının arkaplanını beyaz ve yazı rengini siyaha çevirebilirler:**

Console.BackgroundColor = ConsoleColor.White;

Console.Clear();

Console.ForegroundColor = ConsoleColor.Black;

Console.WriteLine("Merhaba");