**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 Proje2 Yükle 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 (MüşteriAdı, ÜrünSayısı) 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, (Müşteri sınıfı nesneleri içeren) **Generic List** olmasına dikkat ediniz. Öncelikle **Müşteri sınıfını** (MüşteriAdı, ÜrünSayısı sahalarını içeren) oluşturunuz.

**ArrayList içindeki Generic List’lerin her birinin kapasiteleri random sayı (1 ile 5 arasında) attırılarak belirlenecektir ve o kadar eleman alabileceklerdir. Dizilerde eleman kaldıysa sonraki ArrayList elemanı eklenerek devam edilmelidir. A maddesinde yazacağınız metot, herhangi bir dizi boyutu için çözüm üretilebilmelidir. Örnek olarak MüşteriAdı ve ÜrünSayısı dizileri 50 elemanlı ise onun için de çözüm üreten bir metot yazılmalıdır.** MüşteriAdı ve ÜrünSayısı dizilerinin aynı uzunlukta verildiği varsayılabilir.

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

MüşteriAdı = { Ali, Merve, Veli, Gülay, Okan, Zekiye, Kemal, Banu, İlker, Songül, Nuri, Deniz }

ÜrünSayısı = { 8, 11, 16, 5, 15, 14, 19, 3, 18, 17, 13, 15 }

**Örnek : 12 müşteri varsa, attırılan random sayılar sırası ile 4, 2, 5, 1 gelmiş ise Şekil 1’deki yapı ortaya çıkmalı:**

Müşteri

Nuri, 13

Generic Lists

Merve, 11

Veli, 16

Gülay, 5

Songül, 17

İlker, 18

Banu, 3

Zekiye, 14

Ali, 8

Deniz, 15

Kemal, 19

Okan, 15

ArrayList

0

1

2

3

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

**c) Dinamik Dizinizde (ArrayList) kaç tane eleman (Liste) olduğunu buldurup yazdırınız (Şekil 1’deki örnek için 4). Listelerin ortalama eleman sayısını (Müşteri sınıfı nesnesi sayısı / ArrayList’in eleman sayısı) buldurup ekrana yazdıran kodu yazınız (Şekil 1 için 12 / 4). Rapora ortalama eleman 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. Müşteri (sınıfı nesnelerinden oluşan) Yığıtı oluşturacak şekilde kodu güncelleyiniz veya kendiniz yazınız. Soru 1’deki tüm müşterileri, oluşturduğunuz Yığıta ekleyiniz (bilgileri dizilerden de, oluşturduğunuz ArrayList’ten de çekebilirsiniz, size kalmış). Yığıttaki tüm elemanları çıkartıp ekrana müşteri bilgilerini (Müşteri Adı ve sepetindeki ürün sayısı) 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 **sepetindeki ürün sayısı en büyük** olan elemanı arayarak onu silecek). **Müşteri 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, elemanları ürün 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. Bkz. Java kodu: <http://yzgrafik.ege.edu.tr/~ugur/10_11_Fall/DS/DS2010MT_ANSWERS.pdf> (Kod 3)

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

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

**4) (20 puan)**

**a)** Tasarladığınız **ÖncelikliKuyruk** sınıfı ve metotlarını artan sırada işleyecek şekilde güncelleyiniz. 3. sorudaki sınıfınızı kaybetmemek için yedekleyiniz.

**b) Kuyruk** (FIFO yapısındaki) **ve ÖncelikliKuyruk** sınıflarını kullanarak veri yapılarındaki her bir müşterinin işlem tamamlanma sürelerini ve tek kasa için **ortalama işlem tamamlanma** süresini bulunuz, sonuçları rapora ekleyiniz ve kısaca karşılaştırınız. [Müşterilerin kasadaki işlem sürelerinin Ürün Sayısı olduğu varsayılacaktır.]

**c)** Öncelikli Kuyruk (PQ) düzeni, ortalama işlem tamamlanma süresi açısından daha verimli olmaktadır. Ancak dezavantajları nelerdir? Hangi durumlarda kullanılamaz. Kısaca açıklayınız.

**d)** Müşterilerin geliş sırasını da dikkate alan ancak ortalama işlem tamamlanma süresini de azaltacak başka bir yöntem öneriniz (iyi taraflarını birleştiren).

**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, listelerdeki ortalama eleman 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 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");