

CMPE 242

Programlama Ödev 1

Bu Ödev, 4 Nisan Pazar günü saat 23:55'e kadar sona erecek

HW ile ilgili sorularınızı sorabilirsiniz. Lütfen aşağıdaki seçeneklerden birini kullanın:

1.

Moodle Ödev **soru Forumu**: Moodle HW Soru-Cevap (Q&A) Forum

her zaman mevcuttur. Kurs Moodle sayfasındaki "Forum" bağlantısını kullanın.

2.

Ödev **TİLAVETİ saatleri**: takip eden günlerde iki soru-cevap TİLAVETİ saati olacaktır:

•

CMPE242-HW1-OfficeHour1:

Mar 22, 06: 00-07: 00 pm, Zoom kimliği: 952 6936 0016

•

CMPE242-HW1-OfficeHour2:

Mar 26, 06: 00-07: 00 PM, Zoom ID: 928 5510 3025

Not: lütfen katılmadan önce HW belgesini okuduğunuzdan emin olun.

BÖLÜM I

Adım 1

İlk bölüm için, bağlantılı bir liste kullanarak genel ADT yığını uygulamamız gerekir.

Kendi bağlantılı liste uygulamanızı sıfırdan uygulamanız gerekir, yerleşik Java

bağlantılı liste sınıfını kullanamazsınız. Yığın uygulamanızı **myStack** olarak adlandırın ve sınıf

aşağıdaki işlemleri uygulamalıdır:

Boolean isEmpty (), yalnızca yığın boşsa "true" değerini döndürür.

int boyutu()

Yığındaki öğe sayısını döndürür.

void push (öge)

Yığının üstündeki bir öğeyi iter.

Öge pop()

Üst öğeyi yığının üstünden kaldırır ve döndürür.

Öge peek()

En son eklenen öğeyi kaldırmadan yığına döndürür

.

print()

Yığının öğelerini yukarıdan aşağıya doğru yazdırır.

NOT 1: tüm yöntemleri ve tüm istisnaları da test ettiğinizden emin olun.

NOT 2: sınıfı uygulamadan önce jenerik kavramını anladığınızdan emin olun.

2. adım:

Bu adımda,

covid-19 aşısı stokunu yönetmek için yukarıda oluşturduğunuz myStack sınıfını kullanacak bir

VaccineStock uygulaması yazacaksınız. (Not: normalde

gerçek hayatta bu amaç için bir yığın kullanmazsınız, ancak yığınların kullanımını deneyimlemenizi isteriz.)

Önce VaccineStock adlı bir sınıf **oluşturacaksınız**. Bu sınıf üç özel veri üyesine ve üç işleve sahip olacaktır:

seri numarası

Aşı seri numarasını içeren bir tamsayı.

countryName

Aşının yapıldığı ülkenin adını taşıyan bir dize

.

aşıların sayısı

Üretilen aşı sayısını tutan bir tamsayı.

void popitem()

Öğeyi yığından çıkarır ve görüntüler.

geçersiz pushitem()

öğeyi yığına iter

int action()

Eylem menüsünü görüntüler ve kullanıcının seçimini döndürür

Bu program **sonsuz bir döngüde ADD/DELETE/EXIT** komutunu girmek için sorulmalıdır.

Kullanıcı

ekle seçeneğini girerse, program envanter mystack'e yeni bir öge ekler; girdiyi **SİL'DE envanter mystack'ten**

bir ögeyi kaldırın . Kullanıcı EXIT yazana kadar döngü devam etmelidir .

Bir öge eklerken, program kullanıcıya

VaccineStock sınıfının üç veri üyesi için ihtiyaç duyduğu bilgileri sormalı ve yığına yeni bir öge eklemelidir. Bir

ögeyi yığından kaldırırken, program **yığından attı VaccineStock nesnesindeki tüm bilgileri görüntülemelidir.**

Program sona erdiğinde, yığının kalan tüm ögelerini açmalı

ve verilerini görüntülemelidir. Aşağıda örnek çıktı biçimini görebilirsiniz:

```
$ java VaccineStock
```

```
Enter komutu?
```

```
Ekle
```

```
Öge verilerini girin?
```

```
191
```

```
Çin
```

```
15000
```

```
Enter komutu?
```

```
Ekle
```

```
Öge verilerini girin?
```

```
192
```

```
Türkiye
```

```
8000
```

```
Enter komutu?
```

```
Sil
```

```
192, Türkiye, 8000
```

```
Enter komutu?
```

```
Çıkış
```

```
191, Çin, 15000
```

BÖLÜM II

Bölüm II için, dizi yeniden boyutlandırma yaklaşımını kullanarak sıra ADT uygulamanız gerekir.

4'lük bir dizi boyutu ile başlamalısınız ve sınıfta açıklandığı gibi, gerekirse dizinin boyutu artırılabilir (iki katına çıkarılabilir) veya azaltılabilir (yarıya indirilebilir)

.

Adım 1:

Genel bir sıra sınıfı yazın. Sınıfınızı **myQueue** olarak adlandırın. **MyQueue** sınıfı aşağıdaki işlemleri uygulamalıdır. Kendi uygulamanızı uygulamanız gerektiğini unutmayın.

Boolean isEmpty (), yalnızca sıra boşsa "true" değerini döndürür.

boolean isFull ()

Sıra doluyorsa ve yalnızca "true" değerini döndürür.

int boyutu()

Sıradaki öge sayısını döndürür.

void enqueue()

Öğeyi sıranın arkasına ekler.

Öğeyi dequeue ()

Sıranın önünden bir öğeyi siler.

Öğeyi peek()

Sıradan sıradaki öğeyi döndürür.

int action()

Eylem menüsünü görüntüler ve kullanıcının seçimini döndürür.

void print()

Sıranın öğelerini önden arkaya yazdırır.

NOT 1: tüm yöntemleri ve tüm istisnaları da test ettiğinizden emin olun.

NOT 2: sınıfı uygulamadan önce jenerik kavramını anladığınızdan emin olun.

2. adım:

Şimdi

, üniversitenin sağlık merkezinde COVID-19 aşısını bekleyen öğrenci hattının yönetimini simüle etmek için yukarıda oluşturduğunuz myQueue sınıfını kullanacak bir aşı uygulaması yazın

.

İşte varsayımlar:

•

Sağlık Merkezinin günlük aşılama kapasitesi X'dir (yani, belirli bir günde sadece X kişiyi aşılayabilir

).

•

Öğrenciler, bir adı ve büyüklüğü olan öncelikli risk gruplarında gelirler. Örneğin

, Group1 boyutu 40, Group2 boyutu 30 ...

•

Grup büyüklüğünün sağlık Merkezinin

günlük aşı kapasitesinden (X) daha fazla olmadığı garanti edilir (yani varsayılabilir).

•

Sağlık Merkezi ilk önce en yüksek risk önceliklerine sahip grupları aşılamalıdır.

•

Sağlık Merkezi günlük aşı kapasitesini doldurmalıdır.

Program giriş dosyası COVID19'DUR.txt. Dosyanın ilk satırı sağlık merkezinin günlük kapasite numarasını içerir. Aşağıdaki satırlar

, grubun adına ve gruptaki öğrenci sayısına karşılık gelen bir ad ve sayı içerir. Gruplar azalan öncelik sırasına göre sıralanır (yani, ilk grup en yüksek önceliğe sahiptir).

Giriş:

Çıktı aşağıdaki biçime sahip olmalıdır: her gün için,

o gün aşıl原因an grupların adı ve büyüklüğünün bir listesi ile. Örnek çıktı biçimine bakın. Son olarak, aşıl原因an toplam

öğrenci sayısını yazdırın.

Çıkış:

50

Grup1 30

Group2 10

Group3 50

Group4 20

Gün1: Grup1 30 Grup2 10

2. Gün: Grup3 50

3. Gün: Group4 20

Toplam Öğrenci: 110

Toplam Gün: 3