



Fenerbahçe Üniversitesi

BLM 102 – Algoritmalar ve Programlama II

LAB 8: İstisna İdaresi (Exception Handling)

Önemli Not: Aşamaları tamamladıkça, dersin hocası veya asistanı yanınıza çağırarak, tamamladığınız aşamayı gösterdikten sonra diğer aşamaya geçiniz.

LAB Hakkında:

C++ dilinde istisna idaresi kullanımı ile ilgili deneyler yapılacaktır.

LAB'ın aşamaları ve puanlar:

1- Ön aşamalar (0 Puan)

İki sayının toplamını try-catch bloğu içerisinde gerçekleştiriniz.

Sayıların toplamı;

- 10 olduğunda, 20 throw
- 11 olduğunda, 11.5 throw
- 12 olduğunda, "Deneme" throw

etmelidir.

İstenen davranışı gerçekleştirecek try-catch bloklarını yazıp, tüm durumların test edilmesini sağlayınız.

2- Aşama 1 (50 Puan)

Main'de DIZI_BOYUT isminde constant integer bir değişken tanımlayıp başlangıç değerini 50 giriniz. Bu sayıyı kullanarak, sayı kadar genişliğinde bir char dizisi açıp, içindeki elemanları rasgele sayılar ile doldurunuz.

```
void diziGuncelle(char * dizi, int sayi, int indeks, const int diziBoyut);
```

isminde bir fonksiyon tanımlayınız. Bu fonksiyon argüman olarak dizinin başlangıç adresini, güncellenecek sayı, güncellenecek indeks ve dizinin boyutunu almaktadır.

diziBoyut'un değeri indeks'ten daha küçük ve eşit ise, fonksiyon "Dizi dışına erişim!" throw'u atmalıdır.

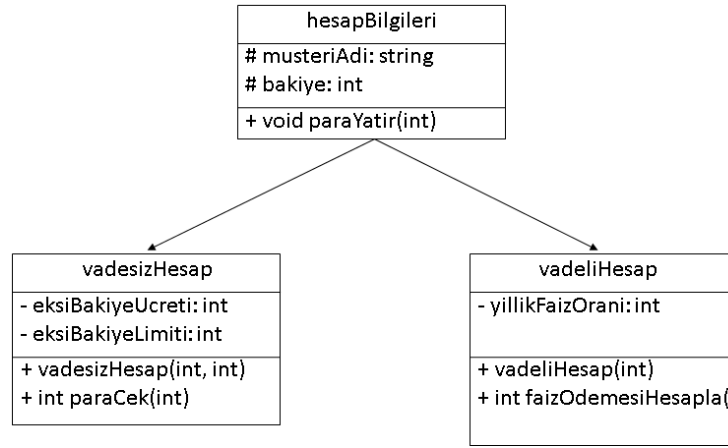
Değilse, indeks pozisyonundaki sayının değerini, argüman olarak aldığı sayı ile değiştirmelidir.

diziGuncelle fonksiyonunu try-catch mekanizması içerisinde test ediniz. İki defa diziGuncelle fonksiyonunu çağırınız. İlk çağırıda argüman olarak (dizi, 3, 5, DIZI_BOYUT) gönderiniz. İkinci çağırıda ise (dizi, 10, 51, DIZI_BOYUT) gönderiniz.

diziGuncelle fonksiyonu çalıştıktan sonra, main'de dizinin değerlerini ekrana bastırınız.

3- Aşama 2 (50 Puan)

Aşağıda birbirleri ile olan ilişkisi UML diyagramı olarak verilen sınıfları geliştiriniz. **Koddaki tüm hata mesajları throw edilerek, catch mekanizmasından ekrana gösterilecektir.**



vadesizHesap ve vadeliHesap sınıfları, hesapBilgileri sınıfından türetilmiştir.

hesapBilgileri sınıfındaki;

paraYatir fonksiyonu, bakiye değişkeninin değerinin üzerine eklenerek güncellemektedir. Aldığı argüman 0'dan küçük ise kullanıcıya hata mesajı göstermelidir.

vadesizHesap sınıfındaki;

Constructor aldığı iki parametreyi, eksibakiyeUcreti ve eksibakiyeLimiti değişkenlerine yazmaktadır.

paraCek fonksiyonu, bakiye değişkeninin değerinden eksilterek güncellemektedir. Aldığı argüman 0'dan küçük ise kullanıcıya hata mesajı göstermelidir. Ayrıca güncellenecek değer en fazla -eksibakiyeLimiti kadar olabilir. -eksibakiyeLimiti'den fazla olduğu durumda kullanıcıya hata mesajı göstermelidir.

Örn. eksibakiyeLimiti 2000 ise, kullanıcının bakiyesi -2000'den daha az olamaz.

Eğer kullanıcının yapılan paraCek fonksiyonu ile bakiyesi 0'ın altına düşüyorsa, bakiyeden ayrıca eksibakiyeUcreti'de çıkartılır.

vadeliHesap sınıfındaki;

Constructor aldığı parametreyi, yıllıkFaizOrani değişkenine yazmaktadır.

faizOdemesiHesapla fonksiyonu çağrıldığında, faizOranına göre bakiye değişkenini artırmaktadır.
Örn faiz oranı 11 ise, $bakiye += bakiye * 11 / 100$ yapılacaktır.