**Öğrencinin Adı, Soyadı: No:**

**NESNE TABANLI PROGRAMLAMA I**

**DÖNEM(QUİZ) ÖDEVİ**

Aşağıdaki banka uygulamasını oluşturunuz.

Bir bankada;

1-Kısa Vadeli

2-Uzun Vadeli,

3-Vadesiz Faizli ve

4-Vadesiz hesap şeklinde dört farklı türden hesap bulunmaktadır ve

her hesap türü aşağıda belirtilen kendine ait Vadesiz Faizliliklere sahiptir.

**Hesap** sınıfı abstract sınıftır.

Bu sınıftan banka hesaplarını temsil eden nesneler oluşturunuz.

**Hesap** sınıfının fonksiyonları:

• mevduat(yatırılan para) : hesaba para yatırılmasına yarar.

• cekmek(çekilen para) : hesaptan para çekilmesine yarar.

Bu işlem yapılırken hesap türüne ve mevcut bakiyeye dikkat edilmelidir.

(Çekilebilecek miktar belirtilmeli, mevcut bakiyeden fazla çekilmeye çalışılırsa hata vermektedir, vb. farklı controller ekleyebilirsiniz.

• getAnlikBakiye : hesabın anlık bakiyesini görüntülemeye yarar.

• getHesapNo : hesap numarasını görüntülemeye yarar. (hesap numarası sabit tam sayıdır).

• Kar(kâr) : hesap açma günü ve işlem tarihine göre kar miktarı hesaplar. (yıllık kar r olursa günlük kar r/365 hesaplanır).

programda 4 tür hesap için tanımlanacak kurallar:

1. KisaVadeli : bu tür hesap yıllık %17 faiz verir ve en az 1000 TL hesapta bakiye olması gerekiyor.

2. UzunVadeli : bu tür hesap yıllık %24 faiz verir ve en az 1500 TL hesapta bakiye olmasi gerekiyor.

3. VadesizFaizli (Vadesiz Faizli Vadeli): bu tür hesap yıllık %12 faiz verir ve en az hesap açtığındakı kadar para hesapta bakiye

olması gerekiyor.

4. Vadesiz: bu hesap faizsizdir ve hesapta para olma zorunluluğu yoktur.

Tüm sistemi kontrol etmesi için başka bir **Banka** adında sınıf tanımlayınız.

Bu sınıfta **Hesap** sınıfı türünden bir pointer olmalı. (**Hesap** \*hesaplar). Bu pointer tüm hesapların adresini tutar.

Bu sınıf şu fonksiyonlara sahip olmalıdır:

1. getHesap : mevcut hesapların listesini görüntülemeye yarar.

2. getTarih: sistemin şu anki tarihini görüntülemeye yarar.

3. mevduat(ID, nakit): ID’si belirtilen hesaba, belirtilen miktarda para yatırma işlemi yapmaya yarar.

4. Cekmek(ID, nakit): eğer mümkün ise, belirtilen miktarda hesaptan para çekme işlemi yapar.

(transaction(işlem) işleminin başarılı ya da başarısız olması gibi durumlar kullanıcıya bildirilir.)

5. cekilis: Vadesiz Faizli hesaplar arasında çekiliş yapmaya yarar. Vadesiz Faizli hesaplarda her 2000 tl için bir puan

tanımlanmalıdır ve çekilişler bu puanlar üzerinden yapılmalıdır. (Yüksek puana sahip olan hesabın

çıkma olasılığı yüksektir.) Kurada çekilen hesap için 10000 TL ödül, hesabına yatırılır.

main()’de **Banka** sınıfından bir nesne oluşturunuz.

Kullanıcı aşağıdaki işlemleri yapabilecektir:

1. Olustur\_S\_ID\_baslangicParasi: Kısa vadeli hesap açar. (hesap no=ID, başlangıç parası = baslangicParasi)

2. Olustur\_L\_ID\_baslangicParasi: Uzun vadeli hesap açar. (hesap no=ID, başlangıç parası = baslangicParasi)

3. Olustur\_O\_ID\_baslangicParasi: Vadesiz Faizli hesap açar. (hesap no=ID, başlangıç parası = baslangicParasi)

4. Olustur\_C\_ID\_baslangicParasi: Cari hesap açar. (hesap no=ID, başlangıç parası = baslangicParasi)

5. Artir\_ID\_nakit: ID’ye ait hesaba “nakit” kadar para ekler.

6. Azalt\_ID\_nakit: ID’ye ait hesaptan “nakit” kadar para çeker.

7. Set\_dd\_mm\_yy: Sistemin şu anki tarihini düzenler.

8. HesabiGoster: Tüm hesapların ID’sini ve yaptığı son 5 işlemi gösterir.

9. GosterIDs: Sistemdeki tüm hesap numaralarını listeler.

10. Sortition: Vadesiz Faizli hesaplar için kura çeker ve sonucu gösterir.

Ek: Sistemin başlangiç tarihi 01.01.2021 alınız. Sınıflara istediğiniz değişken ve fonksiyonlar ekleyebilirsiniz. Hesap bakiyeler için tam sayı kullanabilirsiniz.