



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

2.Ödev Raporu

Grup Elemanları:

B181210110 - Sedanur SARIKOÇ

B181210074 - Merve SARI

SAKARYA

Nisan, 2020

Kütüphane Yapılması ve Test Edilmesi

Sedanur SARIKOÇ^{a*}, Merve SARI^b^a B181210110 / 1C^b B181210074 / 1C

Özet

Ödevin ilk kısmında Java programlama dilini kullanarak rastgele kişi üretebilecek kütüphane tasarlamamız istendi. Bu kütüphaneyi test etmek için ayrıca bir test projesi oluşturmamız istendi. Kütüphanenin kullanışlı olması açısından sınıflar ve bu sınıflar içerisinde her bir işlem için ayrı metotlar yazarak kütüphanemizi geliştirdik. Bu kütüphaneyi test projemize ekleyerek kütüphanemizi test işlemi gerçekleştirdik. Ödevin ikinci kısmında ise birinci kısımda yazmış olduğumuz programı C programlama diline uyarlamamız ve nesne yönelimli programlamaya benzetim yapmamız istendi. Fonksiyon göstericileri kullanarak nesne yönelimli bir yapı oluşturduk. Pointer türü değişkenler ile bellekten yer tahsis ettik ve sonrasında tahsis ettiğimiz yerleri iade ederek çöp oluşmasını engelledik. Ayrıca her iki kısım için dinamik bir yapı oluşturarak farklı txt dosyaları için oluşabilecek hataları önledik.

© 2020 Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Her hangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: Sınıf, kütüphane, fonksiyon, gösterici.

1. GELİŞTİRİLEN YAZILIM

1.1 Birinci kısım

1.1.1 Rastgele sınıfı

Kurucu fonksiyonda bilgisayardan aldığımız zamanı nanosaniyeye çevirerek deger isimli değişkene atadık. RastgeleSayıUret fonksiyonu ile sözde rassal sayı üretici (Pseudorandom number generator) yöntemini kullanarak rastgele bir değer ürettik. if koşul ifadesi ile bu değeri kontrol ederek sayının negatif olması durumunda sayıyı pozitifte çevirdik. Maximum üretmek istediğimiz sayıya göre deger değişkeninin modunu alarak sonuc ifadesine atadık ve sonuc değerini döndürdük.

1.1.2 KimlikNo sınıfı

Kurucu fonksiyonda Rastgele sınıfını kullanarak nesne oluşturduk. Kimlik numarasının ilk hanesi sıfır olamayacağından bunu do-while döngüsünü kullanarak ilk hane sıfır olmayana kadar deger atadık ve döngü çıkışında kimlik numarası hesaplama algoritmasına göre ilk haneyi tek basamakların toplamına ekledik.

* Sedanur Sarıkoç, B181210110

Mail Adresi: sedanur.sarikoc@ogr.sakarya.edu.tr

Sonrasında for döngüsü içinde kimlik numarasının 2.basamağından 9.basamağına kadar rastgele sayı ataması yaptık. Atadığımız basamak tek ise tekToplam değişkenine çift ise çiftToplam değişkenine atadık. Döngüden sonra kimlik numarası hesaplama algoritmasına göre 10 ve 11. basamakları atadık. KimlikNoOlustur fonksiyonu içinde bir for döngüsü kullandık ve bu döngü ile kimlik numarası için ürettiğimiz sayıları string bir ifadeye atadık ve bu ifadeyi fonksiyonda döndürdük. Kimlik numarası doğrulama işlemi yapmak için KimlikNoDogrula fonksiyonu yazdık. Bu fonksiyon içinde fonksiyona parametre olarak gönderilen string ifadenin bütün karakterlerini int bir diziye atadık. Son basamak hariç diğer basamakların toplamını aldık. İf-else koşul ifadesi yardımıyla bu toplamın 10'a göre modunun son basamağa eşit olup olmadığını kontrol ettirdik ve eşit ise geçerli değilse geçersiz ifadesini fonksiyonda döndürdük.

1.1.3 Telefon sınıfı

Öncelikle private olarak Türkiye'de kullanılan telefon alan kodlarını int türünde bir diziye atadık. Sonrasında kurucu fonksiyon içerisinde telefon numarasının ilk hanesinin sıfır olması gerektiğinden ilk haneye sıfır atadık. İkinci hanesine ise başta tanımladığımız alan kodları dizisinden rastgele fonksiyonumuzu kullanarak dizi içerisinde rastgele bir alan kodu atadık. For döngüsü yardımıyla geriye kalan bütün haneleri rastgele sayı üreterek bunların da atama işlemini gerçekleştirdik. TelefonOlustur fonksiyonu içinde bir for döngüsü kullandık ve bu döngü ile telefon numarası için ürettiğimiz sayıları string bir ifadeye atadık ve bu ifadeyi fonksiyonda döndürdük.

1.1.4 ImeiNo sınıfı

Kurucu fonksiyonda imei kodlarını int türünde bir diziye atadık. İmei numarasının ilk iki basamağına atama yapmak için bu diziyi kullandık. Rastgele sınıfını kullanarak nesne oluşturduk. For döngüsü ile imei numarasının 3.basamağından 14.basamağına kadar rastgele fonksiyonumuzu kullanarak atama yaptık. imei numarası hesaplama algoritmasına göre for döngüsü ile tek basamakların toplamını, çift basamaklar için ise gerekli hesaplamaları yaparak bunları tekToplam ve çiftToplam değişkenlerine atadık. Bu toplamaları kullanarak imei algoritmasına göre son basamağı hesapladık. IMEINoOlustur fonksiyonu içinde bir for döngüsü kullandık ve bu döngü ile imei numarası için ürettiğimiz sayıları string bir ifadeye atadık ve bu ifadeyi fonksiyonda döndürdük. İmei numarasını doğrulama işlemi için IMEIKontrol fonksiyonu yazdık. Bu fonksiyon içinde fonksiyona parametre olarak gönderilen string ifadenin bütün hanelerini for döngüsü ile karakterlere bölüp bu karakterleri önceden tanımladığımız basamaklar dizisine atadık. İf-else koşul ifadesi yardımıyla son basamak hariç olarak tek ve çift basamaklar ile ilgili işlemleri yaptık. Yine koşul ifadesi kullanarak son basamağa göre imei kontrolü yapıp geçerli ise geçerli, değilse geçersiz yazısı döndürdük.

1.1.5 Kisi sınıfı

Sınıf içerisinde öncelikle KimlikNo, Telefon, IMEINo ve Rastgele sınıflarından nesne oluşturduk. Kurucu fonksiyonda try-catch bloğu içerisinde try kısmında "random_isimler.txt" dosyasını okuduk ve dosyadaki isimleri daha önce tanımladığımız Kisi ArrayList'ine atadık. Böylece dinamik bir yapı oluşturduk. KimlikNoDondur fonksiyonunda kimlik nesnesi yardımıyla kimlik numarasını döndürdük. IsimDondur fonksiyonunda rastgele nesnesini kullanarak rastgele sayı üretilip bu sayıyla Kisi listesinden rastgele isim seçtik ve bunu döndürdük. YasDondur fonksiyonunda rastgele nesnesi yardımıyla 0-100 arasında random bir yaş üretilip bunu döndürdük. TelefonDondur fonksiyonunda telefon nesnesi yardımıyla telefon numarasını döndürdük. IMEINoDondur fonksiyonunda ImeiNo nesnesi yardımıyla imei numarasını döndürdük.

1.1.6 RastgeleKisi sınıfı

Kisi sınıfından kişi nesnesi oluşturduk ve RastgeleKisi isimli bir string tanımladık. Kurucu fonksiyonda kişi nesnesi ile kimlik numarası, isim, yaş, telefon numarası ve imei numaraları için gerekli fonksiyonu çağırarak bu fonksiyonlardan dönen değerleri RastgeleKisi isimli string'e atadık. RastgeleKisiOlustur fonksiyonunda RastgeleKisi isimli string ifadeyi döndürdük.

1.1.7 RASTGELEKISIURET_TEST dosyası

Yazdığımız kütüphaneyi test etmek için oluşturduğumuz bu test projesinde öncelikle menü oluşturmak için do-while döngüsü kullandık. Bu döngüye koşul olarak çıkış işleminin seçilmesini verdik. Kullanıcı çıkış işlemini seçtiğinde döngüden çıkılacak ve program sonlandırılacaktır. Öncelikle kullanıcının seçebileceği işlemleri ekrana yazdırdık. Bu seçimlerin sonucunda yapılacak işlemleri switch-case bloğu içerisinde gerçekleştirdik. İlk seçenek olan "Rastgele Kişi Üret" ile ilgili işlemleri case 1 içerisinde yaptık. Kullanıcı bu seçeneği seçtiğinde kaç kişi oluşturmak istediğini sorduk ve girilen değeri KisiSayı değişkenine atadık. For döngüsü kullanarak istenen kişi sayısı kadar kişiyi rastgele oluşturduk ve bu kişilerle ilgili bilgileri "Kisiler.txt" isimli dosyaya yazdırdık ve dosyayı kapattık. İkinci seçenek olan "Üretilmiş Dosya Kontrol Et" ile ilgili işlemleri case 2 içerisinde yaptık. Try-catch bloğunun try kısmında KimlikNo ve IMEINo sınıflarından nesne oluşturduk. İlk do-while döngüsü ile "Kisiler.txt" dosyasını okuyarak satır sayısını bulup kişi sayısını arttırdık. Bu kişi sayısına göre KimlikNolari ve IMEINolari isimli diziler oluşturduk. İkinci do-while döngüsü yardımıyla "Kisiler.txt" dosyasının satırlarını boşluğa göre ayırarak her bir satırdaki kimlik ve imei numaralarını daha önceden tanımladığımız KimlikNolari ve IMEINolari dizilerine atadık. For döngüsü içerisinde kimlikNo nesnesini kullanarak kimlik numarası kontrolünü yaptık. Döngü sonrasında kaç tanesinin geçerli olup olmadığını adediyle birlikte ekrana yazdırdık. Başka bir for döngüsü içerisinde imeiNo nesnesini kullanarak imei numarası kontrolünü yaptık. Döngü sonrasında bunun da kaç tanesinin geçerli olup olmadığını adedi ile birlikte ekrana yazdırdık. Üçüncü seçenek olan "Çıkış" seçeneğinin seçilmesi ile programı sonlandırdık. Kullanıcı menüden menüde olmayan bir sayı girmesi halinde ise ekrana "Yanlis tercih!" yazısı çıkardık ve tekrar seçim aldık.

1.2 İkinci kısım

Başlık dosyaları içerisinde yapı, değişkenler, metotlar ve fonksiyon göstericileri ile ilgili tanımlamaları yaptık. Kaynak dosyalarında ise fonksiyon göstericileri için fonksiyonların adres atamalarını yaptık ve yoket fonksiyonları ile bellekten alınan yerleri iade ettik.

1.2.1 Rastgele kaynak dosyası

Java'da yazdığımız koddan farklı olarak kurucu fonksiyon olan RastgeleOlustur fonksiyonunda üretilen deger değişkenine atama yaptık. Do-while döngüsü ile önceki değerden farklı olana kadar zaman ataması yapılarak art arda aynı değer üretilmemesini sağladık.

1.2.2 KimlikNo, Telefon, IMEINo, RastgeleKisi, Test kaynak dosyaları

Java'da yazdığımız koddan farklı olarak tanımladığımız pointer tipi değişkenler için bellekten yer tahsis ettik ve bunları işimiz bitince iade ettik. Java'da yazdığımız metotları C'ye uyarladık.

1.2.3 Kisi kaynak dosyası

Tanımladığımız pointer tipi değişkenler için bellekten yer tahsis ettik. Fonksiyonların dışında tanımladığımız Dugum isimli struct ile bağlı liste yapısı oluşturmamız için gerekli Dugum tipi değişkenlerin tanımlamasını yaptık. If-else yapısı kullanarak dosya açma işlemini gerçekleştirdik. Dosya bulunmuyor ise hata kontrolünü yaptık. Dosyanın bulunması durumunda while döngüsü kullanarak dosyayı dosya sonuna kadar okuduk ve her okunan satır için düğüm oluşturarak bağlı liste yapısını kurduk. Ayrıca dosyanın satır sayısını da bu döngü ile tuttuk. Rastgele değer oluşturarak bağlı listede bu değere kadar ilerleyerek listeden rastgele düğüm seçim işlemini gerçekleştirdik. Bu düğümdeki isim değişkenini kisiIsim değişkenimize atayarak rastgele isim elde etmiş olduk. Kimlik numarası, isim, yaş, telefon ve imei numaraları için yazdığımız fonksiyonlar ile oluşturulan değerleri döndürdük.

2. ÇIKTILAR

Eğer dosya bulunamadı ise hata ifadesi ekranda gösterilecektir. Dosya bulunduğu durumda ise dosyadaki kişi sayısı kadar kimlik numarası ve imei numarası ayrıca bunların geçerlilik durumları çıktı olarak elde edilir.

3. SONUÇ

Metin dosyasında bulunan kişilere ait telefonların ve imei numaralarının geçerliliklerinin kontrolü yapılır. Böylece gerçek hayatta sahtecilik vb. suçların önüne geçilebilir.

Referanslar

- [1] Yumusak N., Adak M.F., *Programlama Dillerinin Prensipleri*. 1.Baskı, SeçkinYayıncılık, 2018
- [2] <https://medium.com/@altunah/rassal-say%C4%B1-%C3%BCretimi-random-number-generation-eeacee141b04>