



**T.C.**  
**SAKARYA ÜNİVERSİTESİ**

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

**Java ve C Dilleri ile Rastgele Kişi Üretebilecek**  
**Kütüphane Tasarlama**

**Grup Elemanları:**

**G171210063- Arif Yusuf Yılmaz**

**G171210009- Arif Damar**

**SAKARYA**

**Nisan, 2020**

## Java ve C Dilleri ile Rastgele Kişi Üretebilecek

## Kütüphane Tasarlama

Arif Yusuf/Yılmaz, Arif/Damar

<sup>a</sup> G171210063 / 2A<sup>b</sup> G171210009 / 2A

---

**Özet (Java ve C)**

Her iki programlama dilinde de rastgele kişi üreten bir kütüphane tasarlanması istenmiştir. Üretilen her rastgele kişi rastgele olarak; ödev dosyasında verilen “random\_isimler.txt” metin belgesindeki isim ve soy isimlerden bir isim ve soy isim, 0-100 arasında bir yaş, Türkiye Cumhuriyeti kimlik numarası algoritmasına uyulacak bir şekilde T.C. kimlik numarası, Türkiye’ de ki telefon numaralarına uyumlu bir telefon numarası, İmei algoritmasına uyulacak bir şekilde imei numarası alır. Üretilen bu kişi “Kişiler.txt” dosyasına yazdırılır ve sonrasında T.C. kimlik ve imei numaralarının geçerlilikleri kontrol edilir. Çalıştırılabilir test programında kullanıcıya 3 seçenekli- rastgele kişi üret, üretilen dosyayı kontrol et, çıkış- bir menü sunulur. Seçilen duruma göre kişi üretilebilir ya da üretilen kişilerin daha önce yukarıda belirtilen özellikleri kontrol edilebilir. Çıkış seçeneği seçilene kadar program devam eder.

Genel problemimiz rastgele sayı oluşturacak fonksiyonların kısıtlanmasından dolayı rastgeleliği kendimizin sağlayıp, kullanacağımız algoritmalara enjekte edebilmek ve sonrasında ödev özetinde bahsedilen konulara uygun bir şekilde programları düzgün bir şekilde çalıştırabilmektir.

Probleme yaklaşım biçimimiz; içerisinde zaman tutan metotları kullanarak zamana erişip, her çağrıldığında sürekli değişmesini sağladığımız değişkenler ile bir dizi işlemlere sokup, sonucunda da her çağrıldığında farklı bir sayı üreten dolayısıyla rastgeleliği sağlamış olan bir algoritma geliştirmektir.

© 2017 Sakarya Üniversitesi.

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

Anahtar Kelimeler: Sözde Rastgele Sayı, Luhn Algoritması, T.C. Kimlik No Algoritması

---

## 1. GELİŞTİRİLEN YAZILIM

### 1.1. Java

Öncelikle istenilen minimum sayı kadar sınıf üretilir. Rastgele sınıfında Zaman'a erişmek için hazır bulunan "Date" sınıfından bir nesne üretilir ve milisaniye döndüren metodu çağrılarak her seferinde farklı sayıya ulaşmak için statik olarak tanımladığımız alt alanlarla bir dizi işleme sokulup sonucunda da rastgele olarak bir sayı döndüren metot oluşturulur.

Kişi sınıfı için ödev dosyasında belirtilen **isim, soy isim, yaş, T.C. kimlik numarası** ve **telefon** alt alanları oluşturulur, "Get" ve "Set" metotları yazılır.

Telefon sınıfında telefon numarasını tutan bir dizi, IMEINo referansı ile imeiNo alt alan olarak oluşturulurlar. imeiNo alt için gerekli Get ve Set metotları yazılır. Sınıf içerisinde ilk hanesi kesinlikle 0, ikinci hanesi kesinlikle 5 olan kalan elemanlarını Rastgele sınıfından oluşturulmuş bir nesne ile sayıOlustur metodu çağrılarak alan fakat bu oluşum içerisinde Türkiye' de ki telefon numaralarına uyumlu olmasına dikkat edilen telefonOlustur isimli bir metot bulunur.

İstenilen sınıflara ek olarak dosya okuma ve yazmaların bulunacağı "DosyaIslem" sınıfı oluşturulur. İçerisinde "icerikDosyaOku", dosyaYaz, uretilmisDosyaOku, dosyaTemizle ve Kontrol metotları bulunur. icerikDosyaOku metodunda isim ve soy isimlerin tutulduğu dosya okunur, okunan her isim ve soy isim alt alan olarak oluşturulan **isimler** ve **soyIsimler** dizilerinde tutulur. dosyaYaz metodunda rastgele kişi bilgileri yeni oluşturulacak dosyaya yazılır. uretilmisDosyaOku metodunda üretilen veya dışarıdan verilen "Kisiler.txt" dosyasındaki her bir satır okunur ve split ile ayrılan her bir bilgi "uretilmis" isimli listede tutulur. dosyaTemizle metodunda parametre olarak verilen dosya temizlenir. Kontrol metodunda döngü yardımıyla üretilmiş listesindeki her eleman bir diziye, o dizideki her bir 0.eleman (T.C. kimlik no tutar) tcKimlikNumaraları listesine, her bir 5.eleman (İmei no tutar) imeiNumaraları listesine eklenir. Dosyanın bozuk olduğu durumlar kontrol edilir ve buna göre bozuk satır sayısı tutulur. Oluşturulan tcKimlikNumaraları listesi KimlikNo sınıfında bulunan tcKimlikKontrol metoduna, **imeiNumaraları** listesi IMEINo sınıfında bulunan imeiNoKontrol metoduna parametre olarak verilerek gönderilir.

KimlikNo sınıfında T.C kimlik numarasını tutan bir dizi alt alan olarak oluşturulur. T.C. kimlik numaraları oluşturulması bir algoritmaya bağlı olduğundan bizden de bu algoritmaya uymamız istenmiştir. Bu sebeple sınıf içerisinde 11 haneli T.C kimlik numarasının 10.elemanını yani 9.indisini bulan metot ve 11.elemanını yani 10.indisini bulan metot ve bu metotlara yardımcı metotlar (**ciftIndisToplamıBul**, **tekIndisToplamıBul**, 10.indise kadar olan **tümIndislerinToplamınıBul**) bulunur. Bu metotlar ilk 9 hanesini rastgele nesnesinin ulaştığı **sayıOlustur** metodu yardımıyla alan **tcKimlikOlustur** metodunda çağrılır, ilk hanenin "0" olamayacağı gibi kontroller yapılır ve böylece T.C. kimlik numarası oluşmuş olur. tcKimlikKontrol metodunda ise kendisine gönderilen tcKimlikNumaraları listesi eleman eleman ayrılarak önce formatsal kontroller (**boş olma durumu, boşluk karakterinden oluşma durumu, uzunluğun 11 haneden oluşmaması**) yapılır daha sonra algoritmik hesaplama

kontrolleri daha önce bahsi geçen indis bulma metotları ve yardımcı metotları sayesinde yapılarak numaranın geçerli ya da geçersiz olduğu anlaşılır.

IMEINO sınıfında imei numarasını tutan bir dizi alt alan olarak oluşturulur. İmei numarası oluşumu da benzer bir şekilde algoritmik hesaplamalar sonucu olduğundan sınıf içerisinde 15 haneli imei numarasının son hanesini bulmak adına oluşturulmuş fonksiyonlar (**ciftToplamBul**, **tekToplamBul**, **genelToplamBul**, **sonHaneBul**) bulunur. imeiNoOlustur metodunda ilk 14 hane-ilk 2 hane hariç- rastgele nesnesinin ulaştığı **sayiOlustur** metodu yardımıyla oluşturulur. İlk 2 hane [https://www.wikiwand.com/en/Reporting\\_Body\\_Identifier](https://www.wikiwand.com/en/Reporting_Body_Identifier) adresinde bulunan İmei numarası standartlarına göre olması gereken kombinasyonlarla oluşturulur. Daha sonra ilgili metotlar çağrılarak 15.hanenin atanmasıyla birlikte imei numarası oluşturulmuş olur. imeiNoKontrol metodunda ise kendisine gönderilen **imeiNumaraları** listesi eleman eleman ayrılarak önce formatsal kontroller (**boş olma durumu**, **boşluk karakterinden oluşma durumu**, **uzunluğun 15 haneden oluşmaması**) yapılır daha sonra algoritmik hesaplama kontrolleri daha önce bahsi geçen indis bulma metotları sayesinde yapılarak imei numarasının geçerli ya da geçersiz olduğu anlaşılır.

RastgeleKisi sınıfında Kisi ve DosyaIslem sınıflarından nesne üretilir. Sınıf içerisinde bulunan **kisiÖzellikAta** metodunda DosyaIslem sınıfından üretilen nesne ile **icerikDosyaOku** metodu çağrılarak dosya okunur, üretilen kişi nesnesinin set ve get metotları ile Kisi sınıfındaki her bir alt alan atanmış olur. Sınıf içerisinde bulunan bir diğer metot olan **listeYap** ile tam sayı dizileri listeye döndürülür. Sınıf içerisindeki son metot olan **kişileriDosyayaYazdir** metodu ile parametre olarak alınan dosyaya, listeYap metodu yardımı ile stringe çevirdiği tam sayı dizileri ile birlikte diğer tüm bilgileri sırasıyla T.C. kimlik numarası, isim, soy isim, yaş, telefon numarası, imei numarası şeklinde yazdırılır.

## 1.2. C

Öncelikle C dili nesne yönelimli programlama desteği olan bir dil olmadığı için sınıf yapıları Struct'lar ile benzetim yaparak kurgulanabilir. Sınıfların kurucu metodları ve sınıfların içinde bulunan metodlar manuel olarak oluşturulmalı. Bu sebeple ödevde istenen her sınıf yapısı için bir Struct (yapı) oluşturulur. Bu yapılar içerisinde fonksiyon göstericileriyle fonksiyonları göstererek nesne yönelimli programlamadaki metotlar taklit edilir.

C programında da Java içerisinde kullanılan çoğu yapı aynı isimle, aynı işlemleri yapacak şekilde bulunur. Ancak bazı işlemler yapılırken Java'da kullanılan yapılar (ArrayList, String, String[]) olmadığından farklı yaklaşımlarda bulunulmuştur. Örnek olarak char şeklinde bulunan bir rakamı integer karşılığına çeviren bir metod bulunmadığı için bu işlem manuel olarak yapılmıştır. Bunun yanında C dilinde string olmadığı ve bunun yerine char\* kullanıldığı için + operatörü birden fazla char\*'ı birleştirme işine yaramıyor, bu yüzden bu işlem **strcat** fonksiyonu ile yapılmıştır. C dili Java dilinden farklı olarak bir çöp toplayıcı mekanizmasına sahip değil, bu yüzden C

dilinde heap bellek bölgesinde oluşturulan her şey manuel olarak iade edilmeli. Program, çalışma süresi boyunca bellekte hiç çöp bırakmayacak şekilde tasarlandı.

Uygulamayı C dilinde yazarken Java’da karşılaşmadığımız türlü zorluklarla karşılaştık. Bunların birçoğunu çözmeyi başardık ancak çözemediğimiz ama farkında olduğumuz birkaç hata hala bulunmakta. Öncelikle C dilindeki program rastgele kişiler üretip dosyaya yazarken bu işlem arka arkaya yapıldığında her zaman olmasa da bazen anlamsız karakterler üretip onları dosyaya yazıyor (Çıktılar’da örneği var). Bunun haricinde İmei ve T.C. kontrolü yaparken okunan dosyada T.C. için 11’den, İmei için 15’ten fazla hane bulunması durumunda program tamamen yanlış sonuçlar üretmektedir. Bu sorunun türlü şekillerde önüne geçmeye çalışsak da başarılı olmadık. **Okunan dosyadaki veriler düzgün formatta olduğu sürece T.C. ve İmei’lerin kurallarına göre kontrol edilmesinde herhangi bir sorun yoktur.**

## 2. ÇIKTILAR

### 1- C hatalarında bahsedilen üretilen anlamsız karakterler

```

135 12163074330 Javier Mcgee 54 05450360238 (504503612785653)
136 41872561060 Ashlynn Kidd 83 05458921016 (981490961258148)
137 74581256750 Andre Drake 67 05058925636 (302589450363210)
138 41034381284 Kareem Pineda 47 05369074789 (306909298743074)
139 28034703680 Alonso Lane 94 05367034585 (306903458547041)
140 25854701620 Carter Dickson 66 05054881716 (100541892981655)
141 87470561200 Casey Hurst 78 05547054985 (306305498123631)
142 27814303660 Charly Jackson 77 05380470749 (863276909478325)
143 90143874354 Ayana Solomon 7 05016123874 (300565232963057)
144 87218705400 Joyami Neal 42 05545096121 (069450961298160)
145 14989216178 Cedric Allison 51 05054707250 (106527834509050)
146 83450741822 Lillah Mclean 70 05323610741 (867276963298748)
147 96327894996 Pranav Nolan 41 05063872161 (013478321610340)
148 36906612930 Sarah Wilkinson 19 05340056721 (500549829610111)
149 85490960280 Skyler Kontoya 81 05541096025 (061650747852168)
150 34385290510 Angie Simmons 62 05323814701 (302381470947879)
151 63094789220 Monica Hurley 66 05307290947 (013632181430765)
152 12563094358
153 Lilly Kane 00 05525630947
154 (065450761278540)
155 96529216190
156 Adrien Clark 39 05058129690
157 (106581490765215)
158 47075520820 Samuel Stanley 61 05530345892 (506303418721038)
159 21834097600 Eden Preston 96 05340094087 (306709498727632x)
160 21814703680
161 Ellen Garner 18 05547036185 (918529630745817y)
162 36925814750 Brittany Clements 42 05069252963
163 (017418529630740)
164 54703692512
165 Jada Lindsey 15 05470369258 (106307418529634zy)
166 41852963648
167 Maria Zimmerman 4 05547036925 (061470741852963)
168 74185296350

```

```

1
200
1-Rastgele Kisi Uret
2-Uretilmis Dosya Kontrol Et
3-Cikis
1
500
1-Rastgele Kisi Uret
2-Uretilmis Dosya Kontrol Et
3-Cikis

```

- 2- Tek seferde üretilen 800 adet kimlik numarası içerisinde sadece 5 tane tekrar bulunuyor. (Her zaman bu kadar düşük oranda olmuyor ancak ortalama olarak 1000 tanede 5-10 tekrar yapıyor) Bu oran İmei için daha düşük telefon için daha yüksektir.

- 3- İşaretlenmiş olanlar hatalı olarak verilmiştir. Program doğru bir şekilde geçerli ve geçersiz sayısını hesaplıyor.

The image shows a Windows desktop with two applications open. On the left is a Notepad window titled "Kisisiler.txt - Notepad" displaying a list of names and IDs. On the right is a terminal window titled "Output - TEST (run)" showing the output of a C# program.

**Kisisiler.txt - Notepad**

File Edit Format View Help

14373073738 Benjamin Mueller 66 05309998591 (107055541550519)  
79969715980 Rebekah Walsh 53 05369378573 (307073741397141)  
36213352994 Blake Wagner 84 05016311413 (505707558575130)  
95239119852 Damian Barker 48 05535754317 (011719161576139)  
96355252546 Jamarcus Meyer 55 05530751833 (306777814375618)  
45512755666 Luca Stark 76 05435547310 (103745194781133)  
61938331284 Josephine Briggs 34 05011967354 (509369550195098)  
69958991242 Jackson Howe 42 05393693783 (306777899561918)  
95017705734 Maci Franco 22 05520916311 (499273949314372)  
50771757956 Miles Cuevas 96 05536353853 (108359835523359)  
59097947774 Johnathon Yoder 26 05341431787 (309057765279706)  
14717070928 Rose Chambers 82 05551167336 (104953499767999)  
8955399668 Mckayla Ellison 8 05534109998 (501785754385599)  
91615761882 Aspen Romero 19 05359295763 (911819067778140)  
35165330310 Lea Ray 79 05335679309 (919753477183333)

**Output - TEST (run)**

T.C. Kimlik Kontrol  
12 Gecerli Tc  
3 Gecersiz Tc  
IMEI Kontrol  
13 Gecerli Imei  
2 Gecersiz Imei  
1- Rastgele Kişi Üret  
2- Üretilmiş Dosya Kontrol Et  
3- Çıkış

### 3. SONUÇ

Bu ödev sayesinde rastgele sayıların bilgisayarlar tarafından nasıl üretilebileceğine dair pek çok bilgiye ulaştık. Aslında kullandığımız çoğu rastgele sayı üretmeye yarayan metodun sözde rastgele sayı ürettiğini öğrendik. Bu ödevde bir uygulamayı iki farklı dil ile yaptığımız için diller arasındaki farklılıkları da oldukça iyi bir biçimde kavramış olduk. Java dili nesne yönelimli bir dil olmasına karşın C dilinin böyle bir desteği olmamasına rağmen C dilinde belli ölçüde nesne yönelimli benzetimlerin nasıl yapılabileceğini bir proje üzerinde deneyimlemiş olduk.

### Referanslar

[https://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_strcat.htm](https://www.tutorialspoint.com/c_standard_library/c_function_strcat.htm)

[https://www.wikiwand.com/en/Reporting\\_Body\\_Identifier](https://www.wikiwand.com/en/Reporting_Body_Identifier)

[https://www.wikiwand.com/en/Luhn\\_algorithm](https://www.wikiwand.com/en/Luhn_algorithm)

<https://teknoseyir.com/blog/t-c-kimlik-numaralarinin-algoritmasi>

<https://www.dcode.fr/duplicates-detector>