T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ ÖDEV RAPORU

ÖDEV-1

Grup Elemanları:

[redacted] - **Ömer Can ÇALIŞIR**

[redacted] - **Melih Ensar BARIŞIK**

SAKARYA

Mart, 2020

Programlama Dillerinin Prensipleri Dersi

Counter

Ömer Can/ÇALIŞIR, Melih Ensar/BARIŞIK

[a] **B171210006 - 1/A**

[b] **B181210393 - 1/A**

Özet

Öncelikle ödevde istenen "icerik.txt" isimli bir dosyanın okunup içerisinde bulunan kelimeleri, sesli harfleri, cümleleri, eposta adreslerini ve web adreslerinin kaç adet olduğunun bulunmasıdır. Programın yapımına sınıfları oluşturarak başladık. Yapılacak işlere göre üç class yapısının olacağını öngördük ve test, readfile ve counter isimli sınıfları oluşturduk. Readfile isimli sınıfta ilgili .txt dosyasından verileri çekmeye yarayan bir fonksiyon yazdık. Bu fonksiyonu kullanarak verileri bir değişkene atadık ve gerekli encapsulation işlemlerini yaptıktan sonra Counter isimli sınıfımızın tasarımına geçtik. Burada sesli harf hesabı, cümle sayısı hesabını, mail sayısı hesabını ve web sitesi hesabını yapacak fonksiyonları yazdık. Bu hesapları gerçekleyebilmek için Regular Expression(Regex) denilen yapıyı kullandık ve Counter sınıfının tasarımı bitmiş oldu.

Regex, InputStreamer, BufferReader, Split

Geliştirilen Yazılım

Ödevin gerçeklemesinde 3 adet sınıf yapısı bulunmaktadır. Bunlar ReadFile, Counter ve Test sınıflarıdır.

ReadFile sınıfında amacımız icerik.txt dosyasından verilerin çekilmesi ve daha sonrasında Counter isimli sınıfın içerisindeki regex fonksiyonlarına aktarılmak üzere düzenlenmesidir. Sınıfın içerisinde ReadFile isimli void bir metod bulunmaktadır. Metod gövdesinde icerik.txt dosyasının okunması için FileInputStream isimli sınıf yapısından fStream isimli bir nesne üretilmiştir ve kurucusuna icerik.txt dosyası verilmiştir. Ardından DataInputStream isimli sınıf yapısından dStream adında bir nesne üretilmiş ve kurucusuna fStream nesnesi verilmiştir. Aynı şekilde BufferedReader isimli bir başka sınıf yapısından bReader nesnesi üretilmiş ve kurucusuna, kurucusu dStream nesnesi olan InputStreamReader isimli sınıf yapısı verilmiştir. Ardından satırları okuyan fonksiyon null olana kadar dönen bir while yapısı tanımlanmıştır. Bu fonksiyon null olana kadar okuduğu satırları bir line değişkenine atamaktadır. Döngü gövdesindeyse önceden tanımlanmış data değişkenine dosyadan okunan her bir satır atanmıştır. While döngüsünden çıkıldıktan sonra data değişkeni split fonksiyonu yardımıyla her bir boşluğa göre ayrılmış ve bu parçalar KelimeDizisi denilen bir diziye atanmıştır. Sonrasında dStream, close fonksiyonuyla kapatılmış ve length fonksiyonuyla KelimeDizisinin uzunluğu bulunup diziUzunlugu değişkenine atanmıştır. Bütün bu işlemlerde hata çıkma olasılığı göz önünde bulundurularak try-catch yapısı kullanılmışır. Devamında encapsulation uygulanan KelimeDizisi ve diziUzunlugu değişkenleri için get fonksiyonları yazılmış ve ReadFile sınıfının tasarımı böylelikle bitmiştir.

ReadFile sınıfından sonra Counter sınıfını tasarlamaya başladık. Öncelikle gerekli değişkenleri yazdık ve diziUzunluguni parametre olarak alan bir kurucu metodu yazdık. Burada gerekli atamaları yaptıktan sonra Atama isimli, ReadFile dan gelecek olan diziyi parametre olarak alan bir metod yazdık. Bu metodun gövdesinde diziUzunlugu kadar dönen ve gelenDizi elemanlarını Counter sınıfı içerisinde oluşturduğumuz Kelimeler dizisine aktaran bir for döngüsü yazdık. Ardından sesliHarfSayisiHesapla isimli bir metod yazdık. Sonrasında bu metodun gövdesinde diziUzunlugu kadar dönen bir for döngüsü yazdık. For döngüsü içerisinde Pattern veri yapısında bir değişken oluşturduk ve bu değişkene Pattern sınıfından statik bir metod olan ve parametre olarak sesli harfleri verdiğimiz compile metodundan dönen değeri atadık. Ardından Matcher veri yapısında bir değişken oluşturduk ve bu değişkene Pattern veri yapısından oluşturduğumuz değişkenin matcher metoduna Kelimeler dizisini verdik. Sonrasında Matcher sınıfının find metodunu kullanarak şart olarak sesli harf bulduğunda sesliHarfSayisini artıran bir while döngüsü yazdık. Böylelikle sesli harf sayısını hesaplayan metod bitmiş oldu.

Sonrasında CumleSayisiHesapla isimli bir metod oluşturduk ve çalışma mantığı sesliHarfSayisiHesapla metoduyla aynı olan bir gövde yazdık. Farklı olarak Pattern sınıfının bir metodu olan compile metoduna parametre olarak cümleleri ayıracak olan Regex kodunu verdik. Ardından aynı şekilde mailSayisi metodu içinde Pattern sınıfının compile metoduna mailleri ayıracak olan Regex kodunu verdik ve tamamen aynı çalışma mantığıyla mailSayisini hesapladık. Yine aynı şekilde webSitesiSayisi metodu içinde Pattern sınıfının compile metoduna websitelerini ayıracak olan Regex kodunu verdik ve diğer metodlarla aynı olacak şekilde webSitesiSayisi metodunu bitirmiş olduk. Sonrasında diziUzunlugunu kelimeSayisi değişkenine atayan basit bir KelimeSayisiHesapla metodu yazdık. Sonuçları yazdırmak adına sonuclarıYaz isimli bir metod yazdık ve içerisinde hesapladığımız değerleri ekrana yazacak print fonksiyonlarını kullandık. Böylelikle Counter sınıfının tasarımı bitmiş oldu.

Counter ve ReadFile sınıflarının tasarımlarını bitirdikten sonra bu işlemleri uygulamaya geçirmek adına Test isimli içerisinde uygulamamızı çalıştıracak main metodu bulunan bir sınıf oluşturduk. Main içerisinde ReadFile sınıfından RF isimli bir nesne oluşturduk.

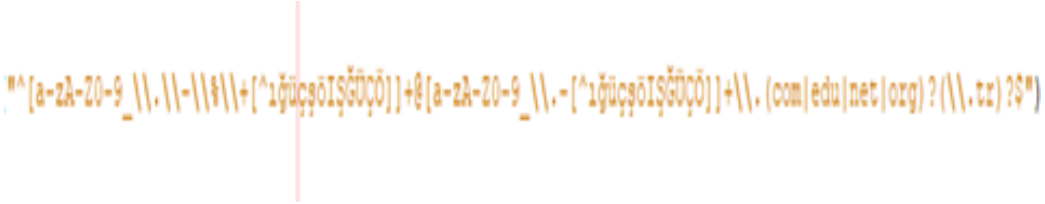
Ardından bu nesneyi kullanarak ReadFile metodunu çalıştırdık ve verileri çekmiş olduk. Devamında Counter sınıfından bir nesne oluşturduk ve kurucusuna getDiziUzunlugu metodunu kullanarak diziUzunlugunu verdik. Atama fonksiyonunu çağırdık ve hesaplama metodlarımızı tek tek çağırdık. Son olarak elde ettiğimiz verileri ekrana yazdırmak üzere sonuclariYaz isimli metodu çağırdık ve uygulamanın tasarımını bitirmiş olduk.

Ödevin verilme amacı Java'daki temel class özelliklerinin kullanması bununla birlikte dosyalama işlemlerinin nasıl yapılacağı olabilir. Verilen işlemler regex metodu kullanılmadan yapılmaya kalkılınca oldukça uğraştırıcı olmaktadır. İnternet üzerinden arama yapmaya yönlendirilip gereken çözümlerin bulunması ve elimizde bulunan probleme uygulanması hedeflenmiş olabilir.

Tablo 1. Metodların Çalışma Durumu

KELİME/CÜMLE	METOD	SONUC
Omer Can	sesliHarfSayisiHesapla	3
Melih	sesliHarfSayisiHesapla	2
Merbaha. Naber?	CumleSayisiHesapla	2

Resim 1. Verilen verinin mail adresi olup olmadığını kontrol eden regex ifadesi.



Algoritma Adımları

1. **Dosyadan verileri çek**
2. **Verileri tek bir string içinde topla**
3. **Toplanan verileri kullanılacak şekle göre ayır**
4. **Ayrılan verileri diziye yerleştir**
5. **Oluşan diziyi kullanılmak üzere diğer sınıfa gönder**
6. **Verilerin içerisindeki sesli harfleri say**
7. **Cümle sayısını hesapla**
8. **Kelime sayısını hesapla**
9. **Mail sayısını hesapla**
10. **Web sitesi sayısını hesapla**
11. **Sonuçları yazdır**

ÇIKTILAR

```
fatihadak@sakarya.edu.tr sakarya.edu.tr www.google.com Esentepe
Kampüs sakarya.com iletisim@ford
```

```
run:
Toplam Sesli Harf Sayısı :33
Toplam Kelime Sayısı     :7
Toplam Cümle Sayısı      :0
Toplam Mail Sayısı       :1
Toplam Web Sitesi Sayısı :3
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
bu bir deneme yazısıdır. deneme+@gmail.com www.deneme.com.tr
www.deneme.com denemeğ@hotmail.com deneme

 Toplam Sesli Harf Sayısı :33
 Toplam Kelime Sayısı     :9
 Toplam Cümle Sayısı      :1
 Toplam Mail Sayısı       :1
 Toplam Web Sitesi Sayısı :2
 BUILD SUCCESSFUL (total time: 0 seconds)
```

Açıklama kısmında anlattığım gibi program oluşturduğu dizinin içine string koyarken boşlukları baz alarak haraket ediyor böylelikle regex yardımı ile gelen her string ifadesini sorgulayabiliyoruz.

Burada ele alınan her bir kelimede Türkçe de var olan sesli harflerin varlığı kontrol edilerek ilk satırda bulunan otuz üç çıktısına ulaşıyoruz.

Kelime sayısı aslında bizim sahip olduğumuz dizinin uzunluğu olduğu için bu değeri oradan elde ediyoruz ve yedi sonucunu ekrana bastırıyoruz.

Son harfi Türkçe cümle bitişinde kullanılan '.' '!' '?' olan dizi elemanlarını sayarak elimizde bulunan cümle sayısını hesaplıyoruz.

Bir mail adresinin onaylanması için içerisinde sadece İngiliz alfabesi harfleri ver 4 adet özel harf('+' '%' '.' '_') olan ve bu kısımdan sonra '@' işareti ile devam edip ödevde istenilen uygun bitiş kelimesi gelmiş ise bir mail adresi olarak sayılıyor.

Web sitesi hesaplarken de ele alınan dizi elemanının sonunda .com .edu .org. .net olması ve ya bunlara ek olarak .tr ifadesi gelmiş olan kelimeleri web sitesi olarak değerlendirip web sitesi sayısını arttırıyoruz.

SONUÇ

Yaptığımız uygulamanın bileşenleri günümüzde web siteleri, masaüstü uygulamaları, mobil uygulamalar vb. ortamlarda sıkça kullanılmaktadır. Örneğin bir web sitesi veya herhangibir uygulamaya kayıt olunurken mail adresi istenmektedir. İstenen mail adresinin mail olup olmadığının kontrol edilmesi gerekir. Biz de ödevi yaparken mail adresi hesaplama işleminde aynı sorunla, bir string ifadenin mail olup olmadığının tespiti sorunuyla karşılaştık. Aynı şekilde ödevde istenen bir diğer işlemse kelime sayısının hesaplanması işlemiydi. Bu işlemde keza gerçek hayatta sıkça kullanılmaktadır. Hatta şu anda bu satırları yazdığım Word dosyası bile yazdığım her kelimeyi saymakta ve aşağıda bilgilendirme amaçlı sonucu göstermektedir. Özetle uygulamada yaptığımız işlemler günlük hayattaki bir çok süreçte kullanılabilir. Bu çalışmayla birlikte ileride karşılaşılması muhtemel sorunlara bir bakış atmış, ilkel düzeyde de olsa çözümler üretmiş olduk.

Referanslar

https://regex101.com/

https://www.geeksforgeeks.org/regular-expressions-in-java/

https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html