# 

T.C.

CUMHURİYET ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ARANAN KELİMEYİ YOKSA BENZERLERİNİ BULAN PROGRAM

## Öğrenci Adı: bATUHAN ÇAM

## Numarası:2015141057

### *Özet:*

Bu proje de yaptığım program öncelikle kelimeler isimli txt dosyasından 100 tane kelimeyi okur ve bir diziye aktarır. Sonrasında bu diziden alarak bir hash tablosunda tutar. Fakat hash tablosuna almadan önce keylerin çakışmaması için belli işlemler yapılır. Bu işlemlerden sonra kullanıcı tarafından aramak istediği herhangi bir kelime istenir. Bu kelime alındıktan sonra yazdığım metodlar sayesinde kelimenin olup olmadığına bakılır eğer kelime varsa kelime bulundu der eğer yoksa kelimenin benzerlerine bakar.(kelimenin harflerinin yer değiştirmiş hali ve harflerinin eksik olarak girilmiş hali bu dosyada var mı?)Bu işlemi 5 kere yaptıktan sonra yani kullanıcıdan 5 kere kelime istedikten sonra ve sonuçlarını döndürdükten sonra program biter.

***Anahtar Kelimeler:*** Hash tablosu, Arama ,Kelime , Quadratic

*Abstract*

In this project, the program I first made is to read 100 words from the named txt file and transfer them to a sequence. Then it takes this string and holds it in a hash table. But before receiving the hash table, certain operations are performed to prevent key conflicts. After this, any word that the user wants to search for is requested. After this word is retrieved, I check whether there is a word or not, if there is a word, the word is found, or if it is not, the word looks similar to the word (is it in the file with the missing letters and letters of the letters)? Once the word is desired and the results are returned, the program ends.

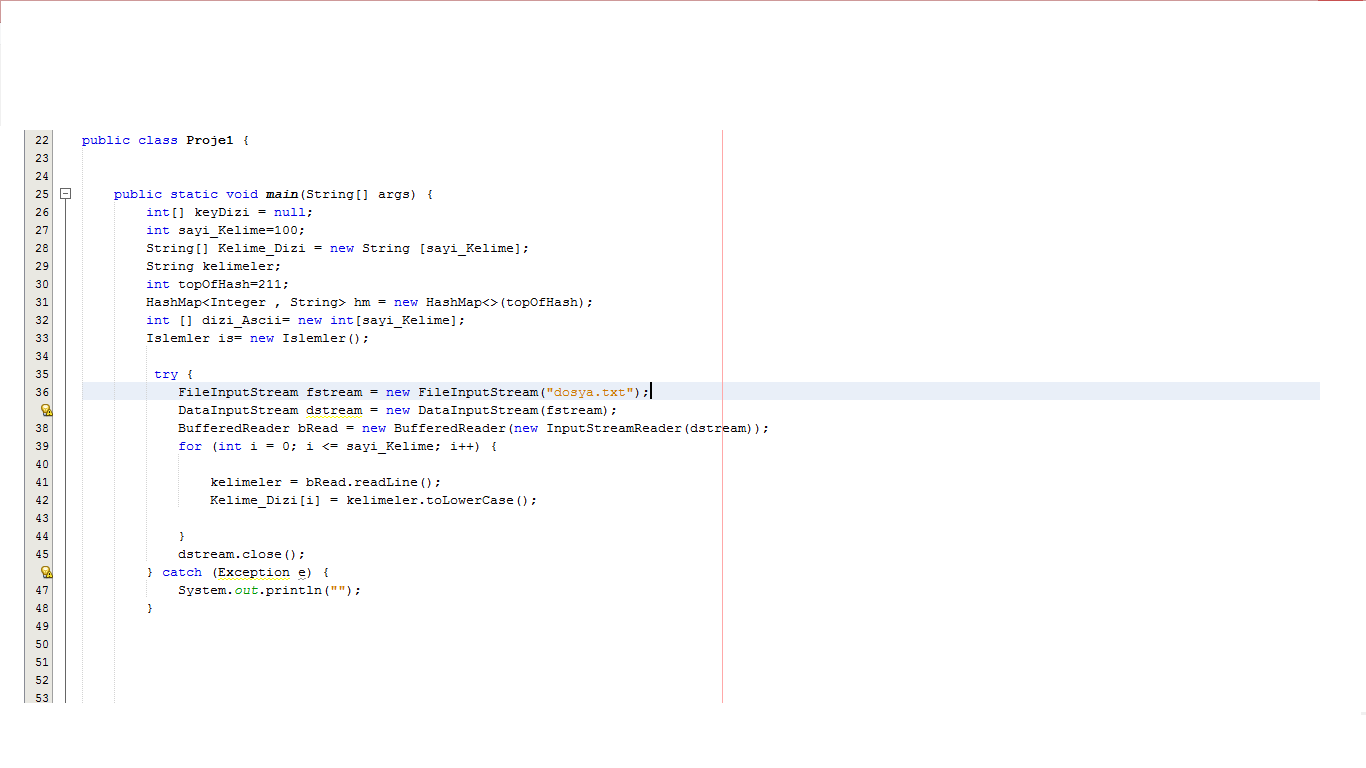
***Keywords:*** Hash table, Searching, Word, Quadratic.

## Giriş

Bu projede txt dosyasındaki kelimeleri programda okutmamız ve onları bir diziye yerleştirmemiz ardından ise bir hash tablosunda tutmamız ve kullanıcının aradığı kelimeyi veya yoksa benzerlerini çıktı olarak vermemiz gerekmektedir. Kelimelerin dosyadan okunduktan ve diziye alındıktan sonra Hash tablosuna yerleştirilirken ascii karsiliklarına göre yerleştirilmesi isteniyor fakat ascii karşılıkları aynı olan kelimeler hash tablosuna key olarak bu ascii karşılıkları girildiğinde aynı olan bu kelimeler tabloda çakışıyor ve aynı yere konuyor.Bunu düzeltmek içinse biz quadratic yöntemi ile bu cakısmayı önlüyoruz.Sonrasında dosyadan bir kelime aranırken eğer varsa direk olarak yazıyoruz fakat bu kelime yoksa kelimenin benzerlerini aramamız gerekiyor.Benzerlikten kastımız ise kelimenin harflerinin yerlerinin değiştirilmiş halleri veya harflerinin eksiltilmiş halleri şeklinde olabilir.Bu sorunları ise yazdığımız metodlarla aynı şekilde aşabiliyoruz.Verilen proje genel olarak bu problemlerle karşılaşır ve bu problemler yazdığımız kodlar ile çözülmüş olur.

## UYGULama

1-Dosya Okuma: Bu kısımda main sınıfında projemizin ilk adımı olan dosya okuma işlemini ve bu okunan kelimeleri bir dizide tutma işlemini gerçekleştiriyoruz. Aynı zamanda diğer metotlarımızda da kullanacağımız değişkenleri tanımlıyoruz buna ilerde değineceğim.



2-Kelimelerin Ascii Karşılıklarını Bulma: Bu metot ise projemizin ikinci adımı olan her kelimenin ascii karşılığını bulur ve bunları bir dizide tutar(dizi\_Ascii).Bu dizi ileri ki metotlar da kullanılacaktır.

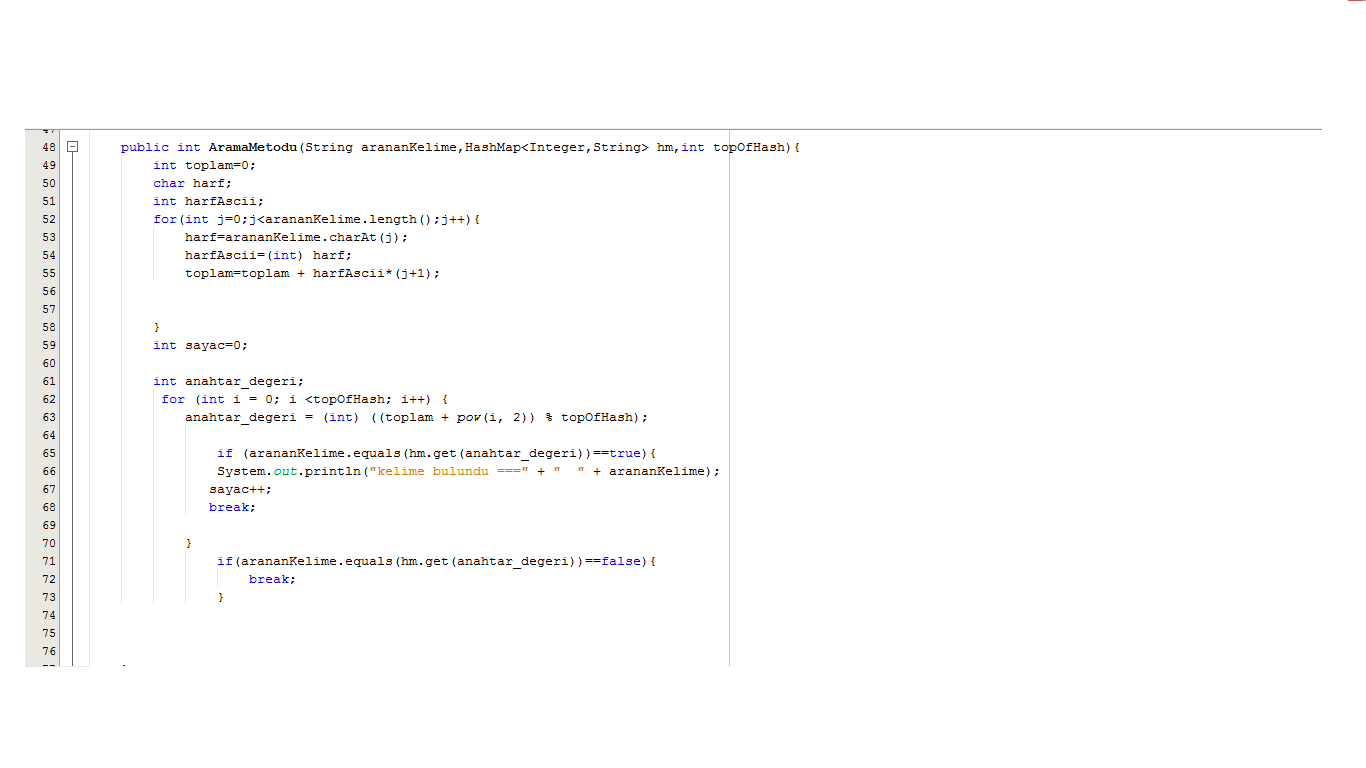


3-Kelimeleri Belirlenen Key Değerlerine Göre Hash Tablosuna Yerleştirme:

Bu metot kelimeleri ascii karşılıklarına göre çakışmayı önleyen quadratic algoritma ile birlikte key değerlerini belirler ve hash tablosuna yerleştirir. Kelimelerin Ascii karşılıklarını bulduğumuz metottaki ascii\_Dizisi adlı dizimizi burada o değerleri almak için kullandık.



4-Arama Metodu: Bu metot hash tablosuna eklenen kelimeler içerisinde kullanıcıdan aranmasını istenen kelime doğrultusunda hash tablosunun keylerine göre bir aramam yapar. Ve sonucu kullanıcıya döner.



5-Kelimenin Yerlerini Değiştirerek Arama Yapan Metot : Bu metot girilen kelime eğer dosyada yoksa o kelimenin benzeri olarak girilen kelimenin harflerinin yerlerini değiştirerek ayrı bir aramam yapar.

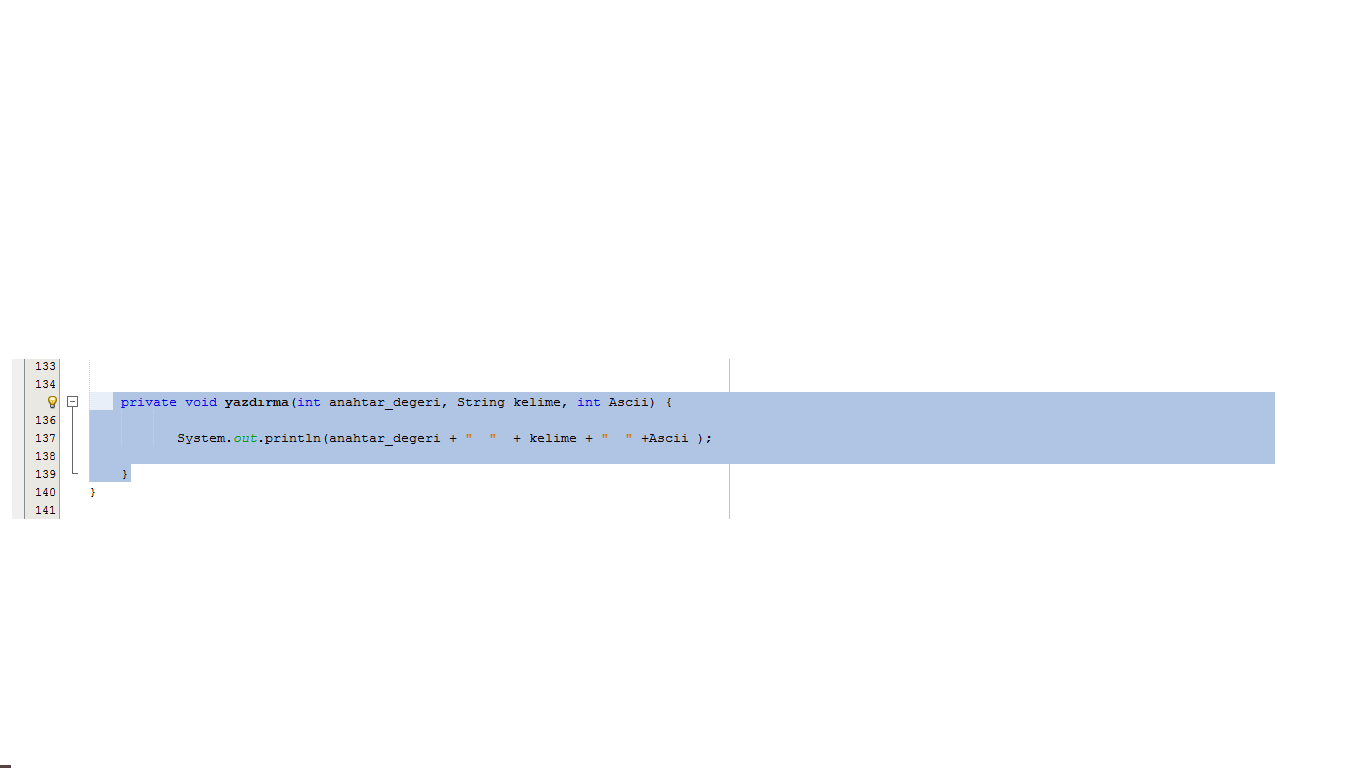
Ve kullanıcıya bu değerleri döner. Bu değerleri dönmesini main sınıfında yapıyoruz.



6-Kelimenin Harflerini Eksilterek Arayan Metot : Bu metot kullanıcının aramak istediği kelime eğer yoksa aranmak istenen kelimelerin harflerini eksilterek bu tür bir benzerliği var mı ona bakar. Ve eğer aranan kelimenin bu tür bir benzeri varsa kullanıcıya bu kelimeleri çıktı olarak verir. Yine bu çıktının verilmesini ana sınıfta yapıyoruz.

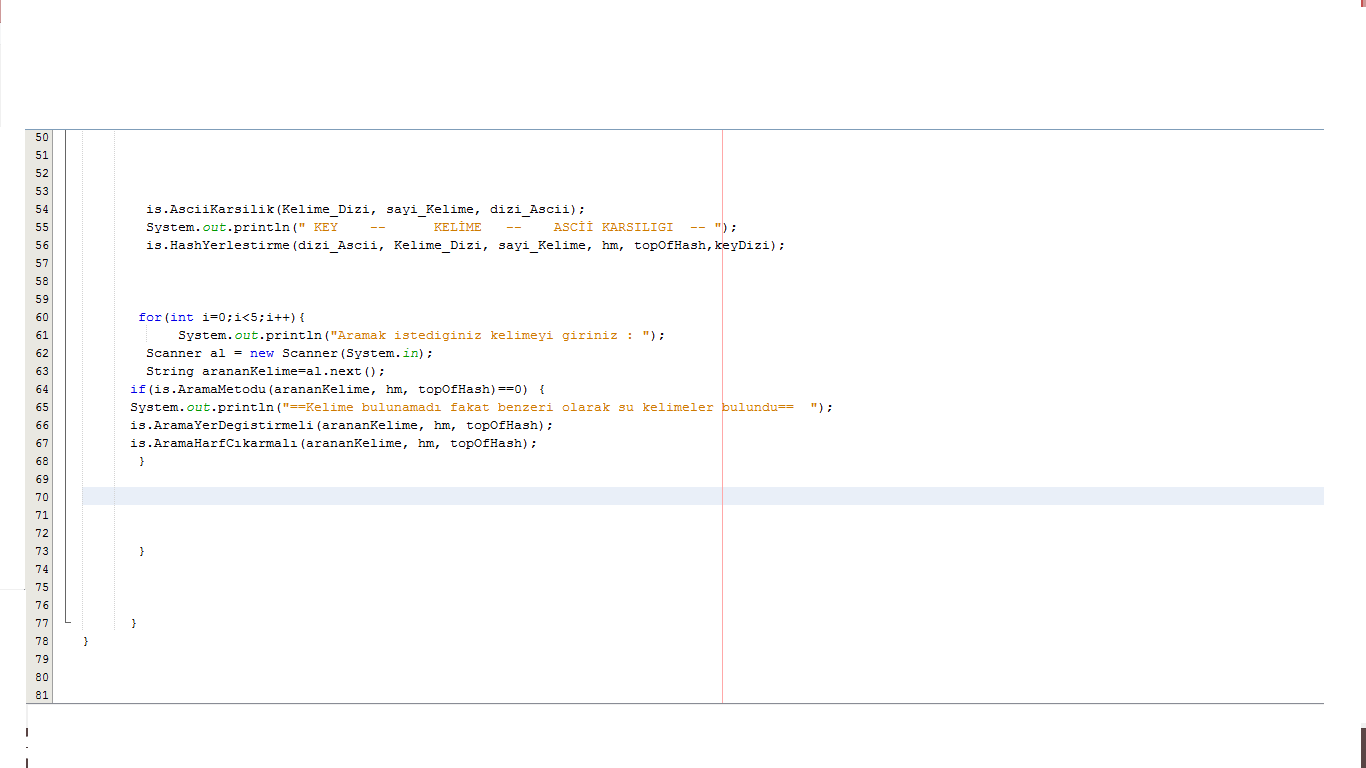


7-Kelimelerin Ekrana Yazılmasını Sağlayan Metot: Bu metot kelimelerin key değerlerini ardından o key değeriyle eşleşen kelimeyi ve kelimenin ascii değerini ekrana yazar. Bu metodu hash tablosuna ekleme yaparken çağırırız detayı sonraki bölümlerde anlatılacaktır.



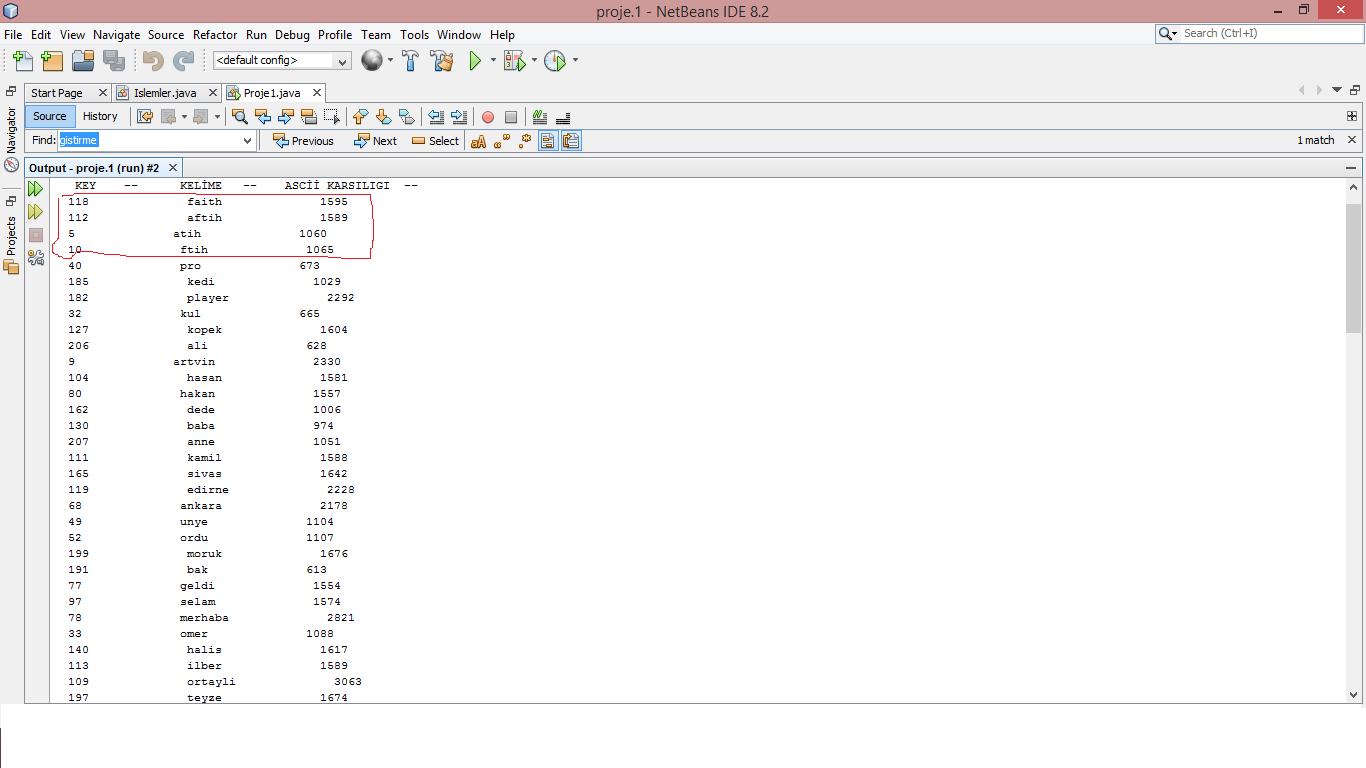
8-Bu kısımda ise ana sınıfımız içerisinde metotlarımızı çağırarak öncelikle kelimelerin ascii karsiliklarini bulan metot ardından hash tablosuna yerleştirdiğimiz metodu çağırıyoruz Sonrasında ise kullanıcıdan aramak istediği kelimeyi girmesini isteyip ardından arama metotlarımızı çağırıp bu kelimeyi arıyoruz.

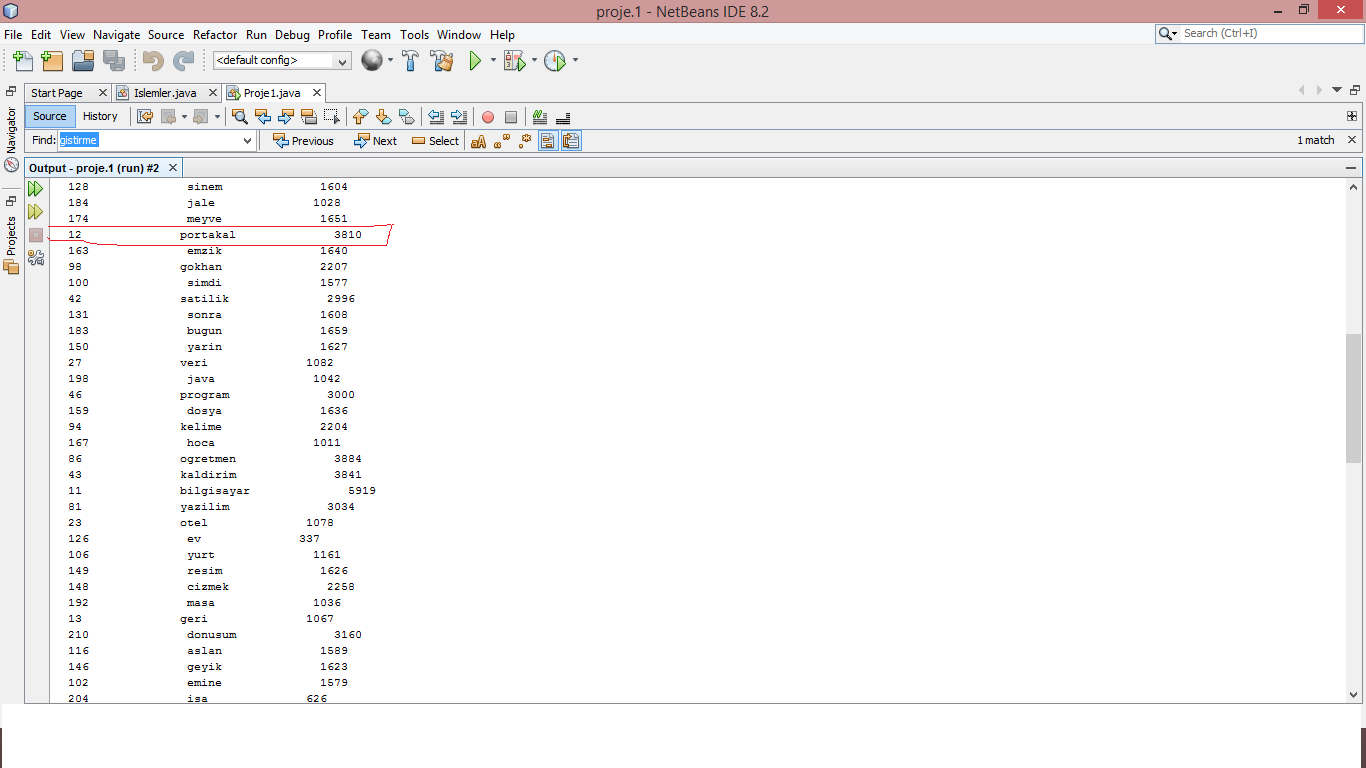
NOT: Ana proje de ana işlemler yapılmıştır. Ascii karşılığı bulma, Hash tablosuna yerleştirme, arama ve yazdırma metotları ise Islemler adlı sınıfta yazilmistir!!

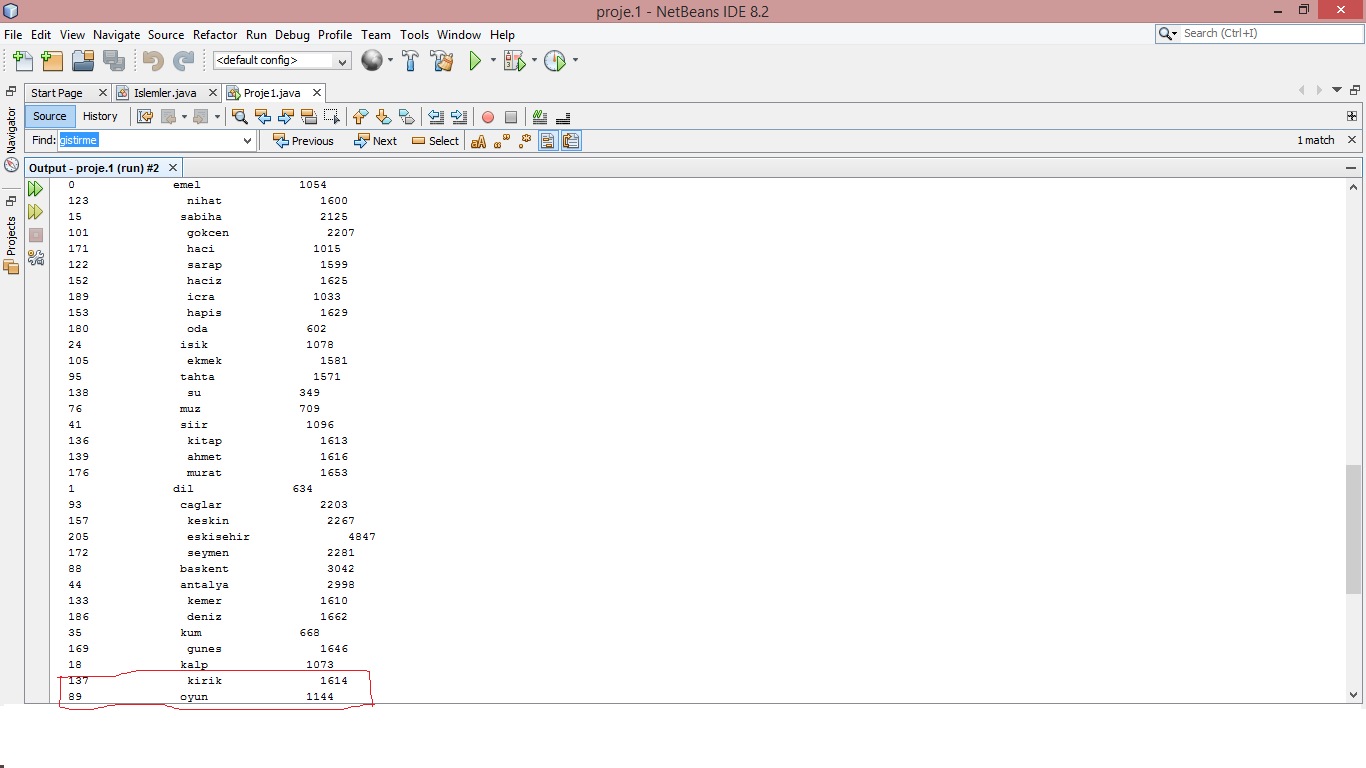


## SONUÇLAR

Öncelikle dosyadaki kelimelerimizi key değerleri ardından kelimeler ve ardından Ascii karşlılıları biçiminde yazdırıyoruz.

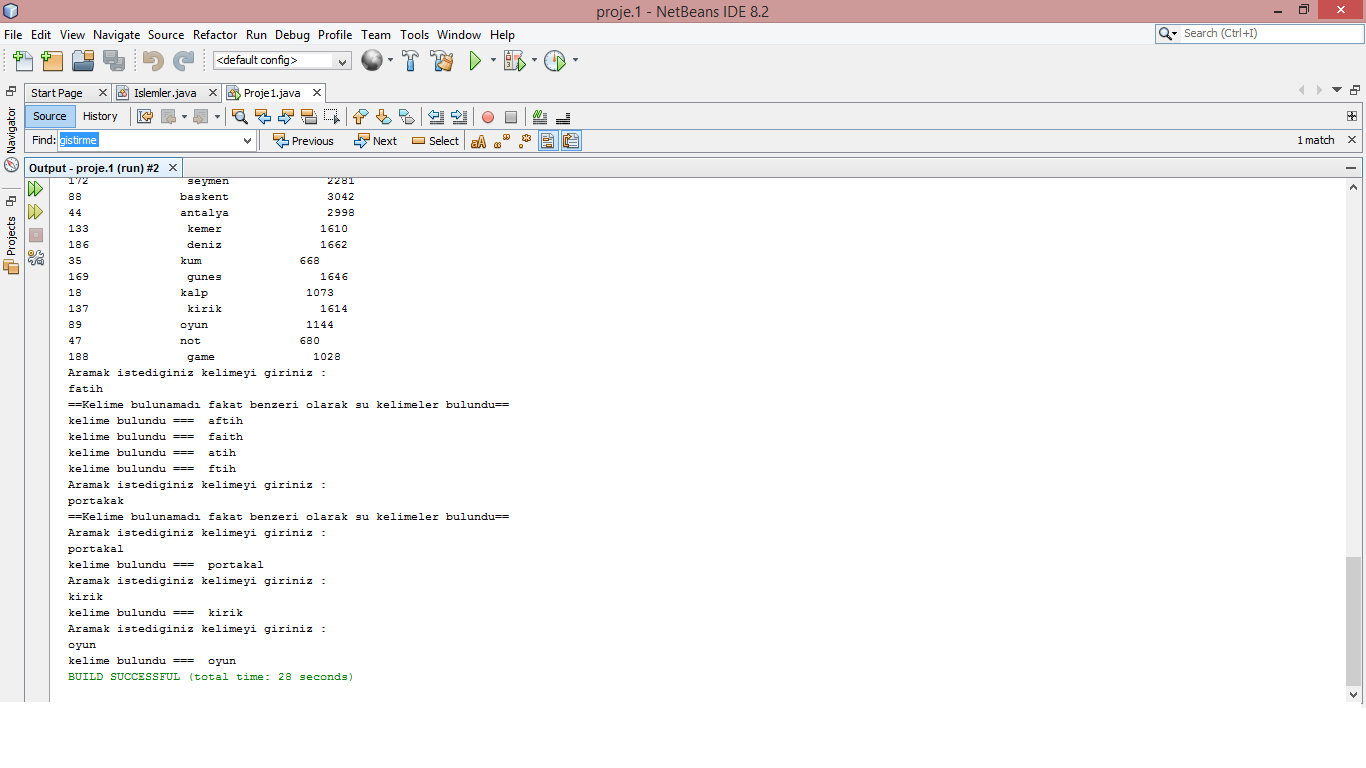






Sonrasında kullanıcıdan aramak istediği kelimeyi isteyip bu kelimelere göre çıktı veriyoruz.

Aranan kelimelerin dosyada olup olmadığı yukarıdaki kısımlarda kırmızı çizgilerle belirtilmiştir.



## KAYNAK KODLAR

package proje.pkg1;

import static java.lang.Math.pow;

import java.util.HashMap;

public class Islemler {

public void AsciiKarsilik(String[] Kelime\_Dizi, int sayi\_Kelime,int [] dizi\_Ascii){//Kelimelerin ascii karşılılarını bulan metot

char harfler;//kelimenin harflerini tutucağımız değişken

int harfinAsciisi;//harflerin ascii karşılılarını tutucağımız değişken

for (int i=0;i<sayi\_Kelime;i++){//bu döngü tüm kelimeleri Words adlı değişkende tutar

String words = Kelime\_Dizi[i];

int toplam=0;

for(int j=0;j<words.length();j++){//yukarıdaki döngünün içinde yazılan bu döngü ise kelime uzunluklarına kadar döner

harfler=words.charAt(j);//ve kelimeyi charAt fonksiyonu yardımıyla harflere böler

harfinAsciisi=(int) harfler;//ardından bu harflerin ascii karşılılarını alır

toplam=toplam + harfinAsciisi\*(j+1);//ve son olarak kelimelerin bütün harflerinin ascii karsiliklarini toplar

dizi\_Ascii[i]=toplam;//ve parametre olarak geçtiğimiz diziye bu asci karşılıları döner

}

}

}

public void HashYerlestirme(int[] dizi\_Ascii, String[] Kelime\_Dizi, int sayi\_Kelime,HashMap<Integer,String> hm ,int topOfHash,int [] keyDizi){//kelimelerini ascii karşılılarına göre hash tobluna yerleştiren metot

int anahtar\_degeri;//quadratic ile bulduğumuz değeri bu değişkene atıp key olarak bu değişkeni kullanacağız

for(int j=0; j<sayi\_Kelime;j++) {

for (int i = 0; i <topOfHash; i++) {

anahtar\_degeri = (int) ((dizi\_Ascii[j] + pow(i, 2)) % topOfHash);quadratic algoritması ile değerlerimizi bulup cakısmayı önlüyoruz ve bu değeri anahtar değeri değişkenimize atıyoruz

if (hm.containsKey(anahtar\_degeri) == false) {//eğer hashimizin keyi bos değilse

hm.put(anahtar\_degeri, Kelime\_Dizi[j]);//kelimelerimizi belirlenen anahtar değerlerine göre hash tablosuna yerleştiriyoruz.

yazdırma(anahtar\_degeri,Kelime\_Dizi[j],dizi\_Ascii[j]);//bu metot ana program içerisinde kelimeleri keyleri ve kelimelerin ascii karşılılarını yazdırır.Tabloya koyduktan sonra yazmayı tercih ettim

break;

}

}

}

}

public int AramaMetodu(String arananKelime,HashMap<Integer,String> hm,int topOfHash){//tablo içerisinde kullanıcının girdiği değere göre arama yapan metot

int toplam=0;

char harf;

int harfAscii;

for(int j=0;j<arananKelime.length();j++){//aranan kelimenin boyutuna kadar sayar

harf=arananKelime.charAt(j);//arananKelime harflerine ayrılıyor

harfAscii=(int) harf;//harflerin ascii karsiliklari alınıp

toplam=toplam + harfAscii\*(j+1);//kelimenin bütün harflerinin ascii karşılıkları toplanıyor

}

int sayac=0;//sayaç ana program içerisindeki if yapısı için konmuştur. If yapısının ne iş yapığı orda anlatılacak.

int anahtar\_degeri;

for (int i = 0; i <topOfHash; i++) {//tekrardan quadratic algoritması ile key değerleri bulunuyor(aranan kelimenin)

anahtar\_degeri = (int) ((toplam + pow(i, 2)) % topOfHash);

if (arananKelime.equals(hm.get(anahtar\_degeri))==true){ //eğer aranan kelimeyi key değeri ile karşılaştırıyor ve eğer true dönerse kelime bulun denilip aranan kelime ekrana yazılıyor.

System.out.println("kelime bulundu ===" + " " + arananKelime);

sayac++;

break;

}

if(arananKelime.equals(hm.get(anahtar\_degeri))==false){//eğer false değer dönerse yoktur demektir hiçbir uyarı yazılmıyor main içerisinde ayarlandı.

break;

}

}

return sayac;

}

public void AramaYerDegistirmeli(String arananKelime,HashMap<Integer,String> hm,int topOfHash)//kelimenin yerlerini değiştirerek arayan metot

char[] YeniArananKelime;//aranan kelimenin karakterlerini tutucagımız dizi

YeniArananKelime=arananKelime.toCharArray();aranan kelimenin karakterleri diziye alınıyor

char gecici;

for(int i=0;i<arananKelime.length()-1;i++){//1 harf sabit kalıp diğerlerinin değişeceğinden dolayı lengt-1.

gecici=YeniArananKelime[i];alttaki 3 işlemde kelimenin harflerinin yerleri değiştirilir

YeniArananKelime[i]=YeniArananKelime[i+1];

YeniArananKelime[i+1]=gecici;

arananKelime=String.valueOf(YeniArananKelime);//bu işlemde ise dizide olan karakterler string olarak tekrar arananKelimeye alınır

AramaMetodu(arananKelime, hm, topOfHash);//aranan kelimenin harflerinin yer değiştirmiş hali arama metoduna yollanarak hash tablosunda var mı diye bakılır

gecici=YeniArananKelime[i];//sonrasında aynı işlemler yapılarak kelime eski haline döndürülür.

YeniArananKelime[i]=YeniArananKelime[i+1];

YeniArananKelime[i+1]=gecici;

arananKelime=String.valueOf(YeniArananKelime);

}

}

public void AramaHarfCıkarmalı(String arananKelime,HashMap<Integer,String> hm,int topOfHash){Aranan kelimenin harflerini çıkararak arayan metot(kaydırma ile)

char [] kelime;

char[] cıkarılanKelime=new char[arananKelime.length()-1];// bu dizi kelimenin harf çıkarıldıktan sonraki halini tutar o yüzden lengt-1 boyutludur

for(int i=0;i<arananKelime.length();i++){

kelime=arananKelime.toCharArray();//arananKelimenin karakterlerini kelime dizisine atıyoruz

for(int j=i;j<arananKelime.length()-1;j++){

kelime[j]=kelime[j+1];//ardından bu karakterleri kaydırıyoruz

}

for(int z=0;z<arananKelime.length()-1;z++){

cıkarılanKelime[z]=kelime[z];//ardından kaydırılmış karakterleri çıkarılan kelime dizisine atıyoruz

}

AramaMetodu(String.copyValueOf(cıkarılanKelime) , hm, topOfHash);//son olarak arama metodunu çağırıp kelime cıkarılmıs halini arama metoduna yollayarak tekrar aranmasını sağlıyoruz

}

}

private void yazdırma(int anahtar\_degeri, String kelime, int Ascii) {//Bu metod program ilk çalıştığı anda kelimelerin key değerlerini kelimeleri ve ascii karsiliklerini ekrana yazmak için yazıldı.

System.out.println(anahtar\_degeri+" "+kelime+" "+Ascii );

}

}

ANA PROGRAM

import com.sun.org.apache.xalan.internal.xsltc.compiler.util.Type;

import static com.sun.org.apache.xalan.internal.xsltc.compiler.util.Type.String;

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.FileInputStream;

import java.io.InputStreamReader;

import static java.lang.Math.pow;

import java.util.HashMap;

import java.util.Scanner;

public class Proje1 {

public static void main(String[] args) {

Ana programın basında kullanacağımız tüm değişkenler tanımlanmıştır.

Ve Islemler sınıfımızın nesnesi oluşturulmuştur.

int[] keyDizi = null;

int sayi\_Kelime=100;

String[] Kelime\_Dizi = new String [sayi\_Kelime];

String kelimeler;

int topOfHash=211;

HashMap<Integer , String> hm = new HashMap<>(topOfHash);

int [] dizi\_Ascii= new int[sayi\_Kelime];

Islemler is= new Islemler();

try {

FileInputStream fstream = new FileInputStream("dosya.txt");

DataInputStream dstream = new DataInputStream(fstream);

BufferedReader bRead = new BufferedReader(new InputStreamReader(dstream));

for (int i = 0; i <= sayi\_Kelime; i++) {//dosya okunur ve kelimeler değişkeninde tutulur ardından burda tutulan kelimeler her seferinde Kelime\_Dizi adlı dizimize harfleri küçültülerek aktarılır.

kelimeler = bRead.readLine();

Kelime\_Dizi[i] = kelimeler.toLowerCase();

}

dstream.close();

} catch (Exception e) {

System.out.println("");

}

is.AsciiKarsilik(Kelime\_Dizi, sayi\_Kelime, dizi\_Ascii);//kelimelerin ascii karşılılarını bulan metot çağırıldı

System.out.println(" KEY -- KELİME -- ASCİİ KARSILIGI -- ");

is.HashYerlestirme(dizi\_Ascii, Kelime\_Dizi, sayi\_Kelime, hm, topOfHash,keyDizi);//ardından hash tablosuna yerleştirildi ve bu metot içinde çağırdığımız yazdırma metodu sayesinde ekrana key değerleri kelimeler ve kelimelerin ascii karşılıları yazıldı.

for(int i=0;i<5;i++){//kullanıcıdan 5 kere kelime girmesini ister

System.out.println("Aramak istediginiz kelimeyi giriniz : ");

Scanner al = new Scanner(System.in);

String arananKelime=al.next();

if(is.AramaMetodu(arananKelime, hm, topOfHash)==0)//aranan kelimeyi alıp arama metoduna yollar ve bu metod içindeki sayaç kelime her bulunduğunda ekrana yazar ve bir artar eğer 0 ise aşağıya geçer. {

System.out.println("==Kelime bulunamadı fakat benzeri olarak su kelimeler bulundu== ")//ve ekrana bu çıktıyı vererek kelimenin yer değiştirmeli ve harf cikarmali arama metodlarını çağırır.Eğer varsa yazar yoksa hicbişey yazmaz.

is.AramaYerDegistirmeli(arananKelime, hm, topOfHash);

is.AramaHarfCıkarmalı(arananKelime, hm, topOfHash);

}

Program 5 kere kelime istendikten ve aramaları yapıldıktan sonra durur.

}

}

}