



**KOCAELİ ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**YAZILIM LABORATUVARI-1  
PROJE -2**

**Aybüke BALIK  
210202107**

**Mehmet Ali AKDOĞAN  
200202017**

## **BÜYÜK VERİDE MULTITHREADING ile BENZER KAYITLARIN TESPİT EDİLMESİ**

### **Özet:**

Bu doküman Yazılım Laboratuvarı I dersinin 2.Projesi (BÜYÜK VERİDE MULTITHREADING ile BENZER KAYITLARIN TESPİT EDİLMESİ)'nin geliştirme ve açıklama niyetiyle oluşturulmuştur. Doküman da projenin isterleri, tanımı, gelişim aşaması , bu süreçte kullanılan araç ve programlar ve en önemlisi kodları paylaşıyor olacağız.

### **Problemin Tanımı:**

Müşteri şikayetleri kayıtlarının tutulduğu bir veri seti içerisindeki benzer kayıtlar tespit edilecek ve tespit edilen kayıtlar masaüstü uygulamasında gösterilecektir. Multithreading kullanarak benzerlik arama süresini düşürmek amaçlanmaktadır

### **2.GİRİŞ**

- Projeyi hazırlarken veri seti içerisindeki arama işlem süresini multithreading kullanılarak azaltmamız, belirtilen sütun/sütunlar için her bir satırdaki kayıtların birbiriyle kelime bazlı karşılaştırılması ve aralarındaki benzerliği tespit etmemiz, uygulama içerisinde istenen özelliklere göre kayıtları filtrelememiz ve kullanıcıya göstermemiz istenmektedir.

### **3.**

#### **İLERLEYİŞ-YÖNTEM**

-Başlamadan Önce: Projeye başlarken thread ve multithread alanlarında araştırmalarımızı yaptık. Swing GUI için uzun süredir yapmadığımız için bu alanda bilgilerimi tazeledik. Elimizden geleni yaptığımızı belirterek araştırma ve raporumuza başlamak istiyoruz.

#### **-Hazırlık:**

Projeye başlamadan önce kullanılacak araç , program ve metodları belirlemek ve iyice anlamak için gerekli bazı tutoriaları inceledik. Hangi dili kullanacağımızı seçmek için araştırmalar yaptık. Arayüz de acemi olduğumuz için Swing GUI tercih ettik. Görev bölümü yaparak projeye başladık.

#### **Kullanılan Araçlar:**

Proje için Java programlama dilini, javada arayüz geliştirmeyi kolaylaştıran swing kütüphanesi kullandık. Bu dili kullanmak için de NetBeans IDE sini ve IntelliJ IDEs sini kullandık.

## -Başlangıç

Projeye önce istenilen verileri filtrelemek için apache csv parser kütüphanesini kullanarak gerekli filtrelemeleri yaptık.

## 4. DENEYSEL SONUÇLAR:

```
1  /*
2
3  List<Tab2> threads = new ArrayList<>();
4
5  long startTime = System.currentTimeMillis();
6
7  for (int i = 0; i < numOfThreads; i++) {
8      Tab2 tab2 = new Tab2( list:nations, incomingData:compareTo, cpg:compare, complaint);
9      threads.add(i, tab2);
10  }
11
12  threads.forEach(thr -> {
13      try {
14          thr.join();
15          thr.interrupt();
16      }
17      catch (InterruptedException e) {
18          throw new RuntimeException("cause:");
19      }
20  });
21
22  long time = System.currentTimeMillis() - startTime; // toplam geçen zaman
23
24  threads.forEach(thr -> {
25      try {
26          model.addElement(thr.getName() + " " + thr.totalTime);
27          lstThreadTime.setModel(model);
28      }
29      catch (Exception e) {
30          e.printStackTrace();
31      }
32  });
33
34  String tm = String.valueOf(time);
35  labelThreadTab2.setText("time:tm");
36  } catch (Exception ex) {
37      ex.printStackTrace();
38  }
39  }
```

Şekil-1 Thread Üretme Kodu ve Geçen Zaman

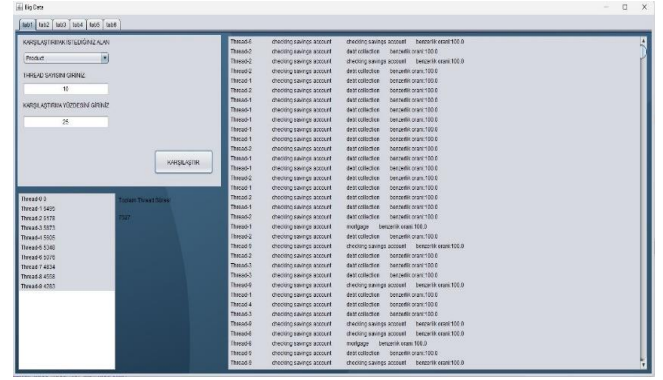
Şekil 1 java da thread üretme ve control etmeye yarayan kod bloğunu kullandık. Bizden istenen thread süreleri olduğu için de sistem saatini kullandık.

```
1 package com.mycompany.swing;
2
3 public class RemovePunctuation {
4     public static String removePunc(String row){
5         row= row.replaceAll( regex:"\\p{Punct}", replacement:"");
6
7         return row;
8     }
9 }
```

Şekil-2 Noktalama İşaretleri Silme

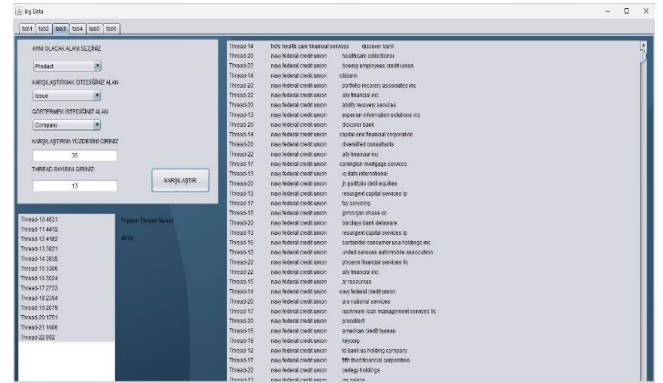
Projenin isterleri içinde bulunan noktalama işaretleri ve özel karakterleri silmemiz sağlayan classı kullandık.

## Program Çıktısı:



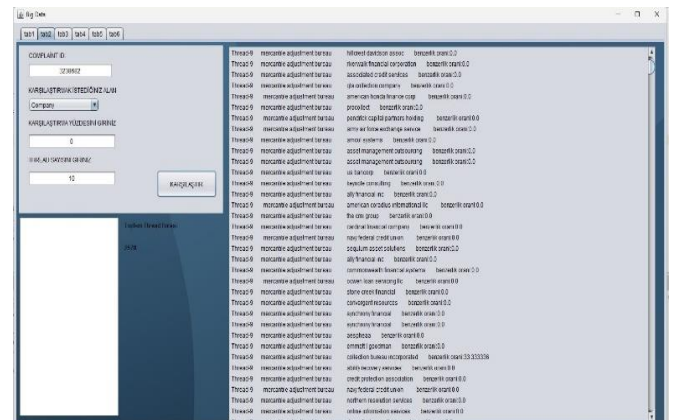
Şekil-3 1.ve 4. Senaryo

Şekil 3'te ise arayüz kısmından kullanıcıdan, karşılaştırmak için sütun, thread sayısı ve karşılaştırma yüzdesi değerlerini aldık ve listeledik.



Şekil-4 2. Senaryo

Şekil 4'te ise arayüz kısmında kullanıcıdan, aynı olacak sütunu, karşılaştırılmak istenen sütunu, ve gösterilmek istenen sütunu, thread sayısı ve benzerlik oranı yüzdesini aldık ve listeledik.



Şekil-5 3. Senaryo

Şekil 5'te ise yine aynı şekilde arayüz kısmında kullanıcıdan, kullanıcının girdiği "Complaint ID" değerine göre karşılaştırılmak istene sütunu thread sayısı ve benzerlik oranı yüzdesini aldık ve listeledik.

## YALANCI KOD

1- BAŞLA

2- CSV DOSYASINI OKU

3- CSV DOSYASINI PARSELLE ve  
"StopWordsRemove" CLASSINDA ve  
"RemovePunction" CLASSRAIYLA FİLTRELE

4-FİLTRELENEN VERİLERİ "nations" ADINDA  
LİSTEYE AT

5- "tab1" ARAYÜZÜNDEN "Thread  
sayısı,karşılaştırma sütunu ve karşılaştırma yüzdesini"  
AL

6- ARAYÜZDEN GİRİLEN KADAR THREAD  
OLUŞTUR

7- "KARŞILAŞTIR BUTONUNA BASILDIĞINDA  
"tab1" CONSTRUCTOR'a GELEN VERİLERİ  
GÖNDER

8- GELEN VERİLERE GÖRE "toCompare"  
FONKSİYONUNDAN ARAYÜZDE  
FİLTRELENMİŞ VERİLERİ VE THREAD  
SÜRELERİNİ GÖSTER

9- İŞLEM BİTTİKTEN SONRA THREADLERİ YOK  
ET

10- "tab2" ARAYÜZÜNDEN "Complaint ID,Thread  
sayısı,karşılaştırma sütunu ve karşılaştırma yüzdesini"  
AL

11- ARAYÜZDEN GİRİLEN KADAR THREAD  
OLUŞTUR

12- "KARŞILAŞTIR BUTONUNA BASILDIĞINDA  
"tab2" CONSTRUCTOR'a GELEN VERİLERİ  
GÖNDER

13- GELEN VERİLERE GÖRE "toCompare"  
FONKSİYONUNDAN ARAYÜZDE  
FİLTRELENMİŞ VERİLERİ VE THREAD  
SÜRELERİNİ GÖSTER

14- İŞLEM BİTTİKTEN SONRA THREADLERİ  
YOK ET

15- "tab3" ARAYÜZÜNDEN "Benzer sütunu,Thread  
sayısı,karşılaştırma sütunu,Gösterilmek istenen sütunu  
ve karşılaştırma yüzdesini" AL

16- ARAYÜZDEN GİRİLEN KADAR THREAD  
OLUŞTUR

17- "KARŞILAŞTIR BUTONUNA BASILDIĞINDA  
"tab3" CONSTRUCTOR'a GELEN VERİLERİ  
GÖNDER

18- GELEN VERİLERE GÖRE "toCompare"  
FONKSİYONUNDAN ARAYÜZDE  
FİLTRELENMİŞ VERİLERİ VE THREAD  
SÜRELERİNİ GÖSTER

19- İŞLEM BİTTİKTEN SONRA THREADLERİ  
YOK ET

20- ÇIKIŞ

## 6.KAYNAKÇA:

Proje geliştirilirken ve araştırma aşamasında  
faydalanılan kaynaklar;

### Web Site

<https://www.javatpoint.com/multithreading-in-java>

[https://www.tutorialspoint.com/operating\\_system/os\\_multi\\_threading.htm](https://www.tutorialspoint.com/operating_system/os_multi_threading.htm)

<https://www.geeksforgeeks.org/multithreading-python-set-1/>

<https://www.mygreatlearning.com/blog/multithreading-in-java/>

<https://www.digitalocean.com/community/tutorials/multithreading-in-java>

<https://www.academia.edu/>

<https://totalview.io/blog/multithreading-multithreaded-applications#:~:text=Multithreading%20is%20a%20model%20of,to%20their%20own%20CPU%20core>

<https://mertmekatronik.com/thread-ve-multithreading-nedir>

<https://www.kaggle.com/datasets/selener/consumer-complaint-database>

<https://www.geeksforgeeks.org/java-swing-jtable/>

<https://www.javatpoint.com/java-jtable>

<https://www.w3schools.com/java/>

<https://www.btkakademi.gov.tr/portal/course/java-ile-programlamaya-giris-9617>