MALATYA TURGUT

ÖZAL ÜNİVERSİTESİ

MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ

Yazılım Mühendisliği Bölümü

YAZM354-Paralel Programlama Dersi Final Ödevi

Hazırlayan: Damla KAYNARCA

Sınıfı:3.sınıf

No:02200201019

Step 1) Yapılacak uygulama kullanıcıda(user) iki tane farklı veri console üzerinden bilgi alınacaktır.Ancak alınacak veriler belli sırada olması gerekmektedir önce birinci thread çalışacak işlem bittiktensonra ikinci thread çalışacaktır. Bunuda Java Paralel programlama özelliğiyle yapılması gerekiyor.Örneğin: 1.thread çalışacak ve sonrasında 2.thread çalışacaktır. (Thread Join özelliğiyle)

Step 2) Birinci thread Kulanıcı bilgileri console üzerindenn Scanner nesnesi üzerinden alınacak username,password ve email adres bunu file bilgisayarın c: \ \ io \ \ turgutozaluniversitesi\\ person.txt dosyasına kaydedilecektir

Step 3) İkinci thread çalışacak kullanıcının secretInformation(tanımlama) isterseniz database kaydedin isterseniz c: \ \ io \ \ turgutozaluniversitesi \ \ secret.txt adındaki dosyaya kaydedin.

Kodlarım

```
import java.io.FileWriter;

import java.io.IOException;

import java.util.Scanner;


public class Main {

public static void main(String[] args) {

// Birinci Thread - Kullanıcı bilgilerini kaydetme

Thread thread1 = new Thread(() -> {

Scanner scanner = new Scanner(System.in);

System.out.print("Kullanıcı Adı: ");

String username = scanner.nextLine();

System.out.print("Şifre: ");

String password = scanner.nextLine();

System.out.print("E-posta Adresi: ");

String email = scanner.nextLine();


try {

FileWriter fileWriter = new FileWriter("c:\\io\\turgutozaluniversitesi\\person.txt");

fileWriter.write("Kullanıcı Adı: " + username + "\n");
```

```
fileWriter.write("Şifre: " + password + "\n");
fileWriter.write("E-posta Adresi: " + email + "\n");
fileWriter.close();
System.out.println("Kullanıcı bilgileri kaydedildi.");
} catch (IOException e) {
e.printStackTrace();
}
});

// İkinci Thread - Secret information kaydetme
Thread thread2 = new Thread(() -> {
Scanner scanner = new Scanner(System.in);
System.out.print("Secret information: ");
String secretInfo = scanner.nextLine();

try {
FileWriter fileWriter = new FileWriter("c:\\io\\turgutozaluniversitesi\\secret.txt");
fileWriter.write("Secret Information: " + secretInfo + "\n");
fileWriter.close();
System.out.println("Secret information kaydedildi.");
} catch (IOException e) {
e.printStackTrace();
}
});

// Threadleri başlatma
thread1.start();
```

```
try {
thread1.join(); // Birinci thread'in işi bitmesini bekler
} catch (InterruptedException e) {
e.printStackTrace();
}
thread2.start();
}}
```

Not: Join() metodoloji birinci thread'in işini bitirmesini beklemek için kullanılır.Böylece ikinci iş parçacığı yalnızca birinci iş parçacığının bitiminden sonra çalışmaya başlar.

Yazdığım kod, bazı kullanıcı bilgilerini ve bazı gizli bilgileri iki farklı metin dosyasına kaydetmek için iki thread oluşturan bir Java programıdır. Kodumda yapılan işlevler:

- Klavyeden giriş okumak için bir Scanner nesnesi oluşturur.
- Aşağıdakileri yapan bir lambda ifadesi çalıştıran bir thread1 nesnesi oluşturur:
- Kullanıcıdan bir kullanıcı adı, bir şifre ve bir e-posta adresi girmesini ister ve bunları klavyeden okur.
- Belirtilen dizinde "person.txt" adlı bir dosyaya yazmak için bir FileWriter nesnesi oluşturur.
- Kullanıcı bilgilerini dosyaya bazı etiketlerle yazar ve dosyayı kapatır.
- Kullanıcı bilgilerinin kaydedildiğine dair bir mesaj yazar.
- Aşağıdakileri yapan başka bir lambda ifadesi çalıştıran bir thread2 nesnesi oluşturur:
- Kullanıcıdan bazı gizli bilgiler girmesini ister ve bunları klavyeden okur.
- Aynı dizinde "secret.txt" adlı başka bir dosyaya yazmak için başka bir FileWriter nesnesi oluşturur.
- Bilgileri dosyaya bir etiketle yazar ve dosyayı kapatır.
- Bilgilerin kaydedildiğine dair bir mesaj yazar.
- Thread1'i başlatır ve join() yöntemiyle bitmesini bekler. Bu, thread1'in thread2'den önce tamamlanmasını sağlar.
- Thread2'yi başlatır ve ana thread ile eşzamanlı olarak çalışmasına izin verir.
- Dosyalara yazarken oluşabilecek herhangi bir IOException'u işlemek için try-catch blokları kullanılır.