

BİL211 - Bilgisayar Programlama II

Lab 5 – Grup2

Bu labda sizden rastgele elemanlardan oluşan listeyi multithreading tekniği kullanarak sıralamanız ve kullanıcı tarafından girilen bir liste üzerinden Java Predicate, Consumer ve Function arayüzlerini kullanarak çeşitli lambda ifadeleri yazmanız bekleniyor.

Dosyalarınızda herhangi bir Türkçe karakter veya package tanımlaması bulunması labdan 0 almanızla sonuçlanacaktır.

Uzak Sistemine Test1.java ve Test2.java olmak üzere 2 java dosyası yüklemeniz beklenmektedir.

Test1.java

```
import java.util.*;
import java.util.function.*;
import java.util.stream.*;

public class Test1 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the list of numbers separated by space: ");
        String input = scanner.nextLine();

        String[] inputArray = input.split("\\s+");
        List<Integer> randomList = new ArrayList<>();
        for (String str : inputArray) {
            int number = Integer.parseInt(str);
            randomList.add(number);
        }

        Predicate<Integer> isOdd =
//verilen integer a sayısının tek olup olmadığını döner, çift ise true tek ise false
        List<Integer> OddNumbers =
// Predicate isOdd ve lambda ifadeleri kullanılarak listeyi yalnızca çift sayıları tutacak
şekilde filtreler. Veriler liste yapısında toplanır.

        Function<Integer, Integer> cube =
// verilen integer a sayısının küpünü döner
        List<Integer> cubes =
// Listedeki her sayı, Function cube ve lambda ifadeleri kullanılarak Küpü ile eşlenir
(map edilir). Veriler liste yapısında toplanır.

        Comparator<Integer> ascendingOrder =
//Verilen a ve b sayısı compareTo metodu kullanılarak kıyaslanır
        List<Integer> A_sortedList =
// Comperator ascendingOrder kullanılarak Liste artan sırayla tekrar sıralanır. Veriler
Liste yapısında toplanır.
```

```

        Comparator<Integer> minNumber =
//Verilen a ve b sayısı compareTo metodu kullanılarak kıyaslanır
        int min =
// Comperator minNumber kullanılarak Liste içerisindeki minimum değeri belirlenir.

        double average =
//liste içerisindeki değerler integer'a çevrilir ve değerlerin ortalaması hesaplanır
(Hint: mapToInt(Integer::intValue))

        System.out.println("Random list: " + randomList);
        System.out.println("Odd numbers: " + OddNumbers);
        System.out.println("Cubes: " + cubes);

        System.out.println("Sorted list Ascending: " + A_sortedList);

        System.out.println("Min number: " + min);
        System.out.println("Average: " + average);
    }
}

```

Test2.java

```

import java.util.*;
import java.util.concurrent.*;
import java.util.stream.*;

public class Test2 {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the size of list: ");
        int N = scanner.nextInt();
        Random random = new Random();
        List<Integer> list = random.ints(N, 1, 100).boxed().collect(Collectors.toList());

        System.out.println("List: " + list);
        System.out.print("Enter the Number of Threads: ");
        int numThreads = scanner.nextInt();
        Thread[] printThreads = createThreadsToPrintSorted(list, numThreads);
        for (int i = 0; i < numThreads; i++) {
            try {
                printThreads[i].join();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

private static Thread[] createThreadsToPrintSorted(List<Integer> list, int numThreads)
{
    /*
    numThread parametresinde belirtilen sayı kadar Thread oluşturulur.
    Her bir thread thread numarasını, (liste büyüklüğü / thread sayısı)
    büyüklüğündeki liste parçalarındaki elemanları ve bu elemanların toplamını
    ekrana bastırır.
    Her bir thread bu fonksiyon içerisinde başlatılır. Fonksiyon oluşturulan
    threadleri bir thread dizisi halinde döner.

    ****liste büyüklüğü thread sayısına tam bölünen bir sayı olacaktır.
    ****Elemanların toplamı için lambda işlemi kullanmanız beklenmektedir.
    ****main thread'i, devam etmeden önce tüm iş parçacıkları bitene
        kadar beklemelidir.
    */
    return threads;
}
}

```

Test1.java

Enter the list of numbers separated by space: 4 1 5 2 3

Random list: [4, 1, 5, 2, 3]

Odd numbers: [1, 5, 3]

Cubes: [64, 1, 125, 8, 27]

Sorted list Ascending: [1, 2, 3, 4, 5]

Min number: 1

Average: 3.0

Test2.java

Enter the size of list: 6

List: [48, 54, 79, 21, 88, 98]

Enter the Number of Threads: 3

Thread 0 printing chunk of sorted list:

Thread 2 printing chunk of sorted list:

Thread 1 printing chunk of sorted list:

[79, 21] Sum: 100

[48, 54] Sum: 102

[88, 98] Sum: 186

NOT: Test2.java dosyasında çıktı sıraları değişebilir.

