

ITB_IF2212_2_2324 Pemrograman Berorientasi Objek STI

[Dashboard](#) / [My courses](#) / [ITB_IF2212_2_2324](#) / [Praktikum 7: Concurrency](#) / [Praktikum 7](#)

Started on

Monday, 6 May 2024, 8:02 AM

State

Finished

Completed on

Monday, 6 May 2024, 9:42 AM

Time taken

1 hour 40 mins

Grade

180.00 out of 300.00 (60%)

Question 1

Correct

Mark 100.00 out of 100.00

Flag question

Time limit

1 s

Memory limit

64 MB

Concurrency - Delayed Output

Lengkapi metode static printDelayed yang mengeluarkan suatu string setelah delay sekian milidetik. Metode printDelayed harus bersifat non-blocking, artinya metode harus bisa dieksekusi lagi, walaupun string yang diekspetasikan belum di print.

Misalnya metode printDelayed dipanggil seperti ini:

DelayedOutput.printDelayed(200, "Halo Obus");
DelayedOutput.printDelayed(100, "Halo Obus 2");

Maka output akan seperti ini:

Halo Obus 2
Halo Obus

Lengkapi dan submit file DelayedOutput.java

Java 8

DelayedOutput.java

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.21 sec, 28.48 MB
2	25	Accepted	0.27 sec, 30.66 MB
3	25	Accepted	0.27 sec, 28.63 MB
4	25	Accepted	0.57 sec, 29.10 MB

Question 2

Partially correct

Mark 80.00 out of 100.00

Flag question

Time limit

1 s

Memory limit

64 MB

Multithread Array Sum

Terdapat sebuah array berisi bilangan bulat. Implementasikan sebuah program untuk menjumlahkan seluruh elemen di dalam array dengan menggunakan java Multithreading.

- Kelas ArraySum memiliki atribut `numWorkers` yang menyatakan jumlah maksimum threads yang tersedia dan `array` untuk menampung array masukan

- Kelas ArraySum memiliki method `sum()` yang akan membuat sejumlah thread dan memetakan array masukan secara merata ke semua threads yang dapat dibuat. Contohnya:

```
...  
array = [1, 2, 3, 4, 5]  
numWorkers = 2  
  
maka:  
Thread-0 <- [1, 2]  
Thread-1 <- [3, 4, 5]  
...  
  
masing-masing thread kemudian memanggil method 'partialSum(int start, int end)' untuk mendapatkan hasil penjumlahan elemen-elemen-nya  
Perhatian:  
Jangan membuat kelas baru untuk membuat suatu thread. Gunakanlah deklarasi in-line kelas, seperti berikut  
...  
  
Thread thread = new Thread(new Runnable() {  
    @Override  
    public void run() {  
    }  
});  
...  

```

- Kelas ArraySum memiliki method `partialSum(int start, int end)` akan melakukan penjumlahan elemen-elemen array pada index `start` sampai `end-1`

Lengkapilah file `ArraySum.java` pada file `zip` ini. Agar tidak membebani server Olympia, gunakanlah file `ArraySumDriver.java` untuk pengujian kompilasi dan solusi awal Anda. Pengujian sebenarnya tetap dengan submit ke Olympia. Kumpulkan file `ArraySum.java` yang telah diimplementasikan.

Java 8

ArraySum.java

Score: 80

Blackbox

Score: 80

Verdict: Wrong answer

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.29 sec, 28.82 MB
2	0	Wrong answer	0.41 sec, 30.25 MB
3	20	Accepted	0.62 sec, 29.97 MB
4	20	Accepted	0.56 sec, 29.01 MB
5	20	Accepted	0.53 sec, 27.86 MB

Question 3

Not answered

Marked out of 100.00

Flag question

Time limit

1 s

Memory limit

64 MB

Word Count Server

****Problem Statement****
Pada kuliah pengembangan aplikasi sistem terdistribusi, Anda akan belajar beberapa jenis web server. Beberapa yang populer yaitu event-driven and thread-based server system. Contoh event-driven webserver yaitu NodeJS dan thread-based yaitu Apache webserver. Pada thread-based webserver setiap request yang datang dari client akan dibuatkan thread baru dan pemrosesan request dilakukan pada thread tersebut.

Disini anda ditugaskan untuk mensimulasikan dan mengimplementasikan simple thread-based webserver yang akan menghitung jumlah kata yang diberikan oleh client.

Server akan menerima kalimat-kalimat yang perlu dihitung jumlah katanya. Kemudian, untuk setiap worker thread yang tersedia pada server akan memproses setiap kalimat tersebut dan pada akhirnya main thread akan mengembalikan jumlah kata pada setiap kalimat.

****Low Level Design****
- Terdapat kelas 'Server' memiliki variabel 'workers' yang menyatakan maksimum jumlah threads yang tersedia pada server (disebut juga sebagai thread pool).
- Method 'process(String[] requests)' akan memetakan setiap elemen pada array requests ke suatu thread secara rata terlebih dahulu. Jika jumlah thread yang tersedia kurang dari jumlah elemen 'requests' maka akan terdapat satu atau lebih thread yang mendapatkan elemen lebih. Contohnya jika 'workers = 3' dan 'requests.length = 4' maka pemetaan elemen:
...
Thread-0 <- requests[0]
Thread-1 <- requests[1]
Thread-2 <- requests[2]
Thread-0 <- requests[3]
...
Terlihat bahwa Thread-0 akan mendapatkan 2 elemen.

Perhatian:
Jangan membuat kelas baru untuk membuat suatu worker thread. Gunakanlah deklarasi in-line kelas, seperti berikut
...

Thread thread = new Thread(new Runnable() {
 @Override
 public void run() {
 }
});
...

- Method 'count(String request)' akan menghitung jumlah kata pada kalimat 'request'. Method ini akan dipanggil oleh setiap worker thread.

Files
- Lengkapilah file 'Server.java'
- Agar tidak membebani server Olympia, gunakanlah file 'ServerDriver.java' untuk pengujian kompilasi dan solusi awal Anda. Pengujian sebenarnya tetap dengan submit ke Olympia.

Submit kembali file 'Server.java' yang telah berisi jawaban Anda.

Java 8

Quiz navigation

1

2

3

Show one page at a time

Finish review

→ [Praktikum 6 - Latihan](#)

Jump to...

[Finish review](#)

[Praktikum 7 - Latihan](#) →

You are logged in as [18222105 Rajendra Farras Rayhan](#) ([Log out](#))
[ITB_IF2212_2_2324](#)
[Data retention summary](#)
[Get the mobile app](#)