

Started on	Saturday, 17 May 2025, 10:06 PM
State	Finished
Completed on	Sunday, 18 May 2025, 4:09 PM
Time taken	18 hours 2 mins
Grade	300.00 out of 300.00 (100%)

Question **1**
Correct
Mark 100.00 out of 100.00

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 Sera"); DelayedOutput.printDelayed(100, "Halo Sera 2");
```

Maka output akan seperti ini:

Halo Sera 2
Halo Sera

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	0.16 sec, 28.92 MB
2	25	Accepted	0.26 sec, 28.93 MB
3	25	Accepted	0.26 sec, 30.00 MB
4	25	Accepted	0.56 sec, 30.15 MB

Time limit	1 s
Memory limit	64 MB

Array Sum 2025

Tiba-tiba kamu dikasih tugas untuk bikin program yang menjumlahkan elemen-elemen array. Terus kamu be like: ini tugas gampang banget, anak SD juga bisa. Terus kamu merasa ga terima, eh ini soal kegampangan. Terus otakmu yang jenius berpikir, gimana kalau kita bikin program ini dengan multithreading? Akhirnya kamu pun berusaha untuk bikin program ini dengan multithreading.

Cerita di atas hanyalah cerita fiksi, tapi soal pra prak ini beneran. Sekarang kamu diminta untuk membuat program yang menjumlahkan elemen-elemen array dengan menggunakan multithreading. Berikut adalah spesifikasi program yang harus kamu buat.

Spesifikasi

Kamu diberikan sebuah array berisi bilangan bulat. Implementasikan sebuah program untuk menjumlahkan seluruh elemen di dalam array. Namun, kali ini kamu diminta untuk menggunakan fitur multithreading dalam Java. Program ini harus dapat membagi tugas penjumlahan ke dalam beberapa thread, sehingga setiap thread hanya menjumlahkan sebagian dari array tersebut.

- Kelas ArraySum memiliki atribut `nWorkers` 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]
nWorkers = 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`

Kumpulkan file `ArraySum.java` yang telah diimplementasikan. Kamu bisa menggunakan `ArraySumDriver.java` untuk menguji program kamu.

Attachments

Attachments: [attachments.zip](#)

Java 8

[ArraySum.java](#)

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.64 sec, 28.73 MB
2	20	Accepted	0.43 sec, 33.08 MB
3	20	Accepted	0.37 sec, 28.09 MB
4	20	Accepted	0.34 sec, 27.91 MB
5	20	Accepted	0.68 sec, 28.31 MB

Time limit	1 s
Memory limit	64 MB

Problem Statement

Pernahkan Anda menulis program yang menjumlahkan semua elemen dalam suatu array? Disini anda ditugaskan untuk menulis program yang sama, tetapi dengan memanfaatkan Java multithreading.

Pada suatu hari, pak Marcel yang sedang gabut, ingin menghitung berapa banyak kata spesial yang merupakan sebuah angka 1,2,3 dalam bahasa inggris yaitu "one", "two" dan "three" yang dapat dibentuk dari list of array yang dia input. Setiap karakter yang sudah dipakai untuk sebuah kata spesial tidak akan bisa dipakai untuk kata spesial lainnya. Misal terdapat sebuah list of string ["one", "tw"], maka hanya terdapat satu kata spesial, yaitu kata "one". Selain dari itu, pak Marcel lebih memprioritaskan kata yang lebih kecil dibanding kata yang lainnya. Sehingga jika pada sebelumnya sebenarnya pak Marcel bisa mendapatkan two, karena terdapat 1 huruf t, 1 huruf w dan 1 huruf o, namun pak Marcel lebih memilih angka one. Kesimpulannya, pak Marcel lebih memilih kata "one" dibanding "two" dan kata "two" dibanding "three". Tugas anda adalah membuat fungsi yang dapat mengembalikan sebuah map berisi berapa banyak kata spesial "one", "two" dan "three" dengan prioritas yang telah dijelaskan.

Low Level Design

- Terdapat kelas `ComplexWordCountServer` memiliki variabel `nWorkers` yang menyatakan maksimum jumlah threads yang dapat dibuat
- Method `countSpecialString(string str)` akan membuat thread dan memetakan array list of string masukan secara rata ke semua threads yang dapat dibuat. Contohnya:

```
array = ["oneone", "two"]
nWorkers = 2

maka:
Thread-0 <- "oneone"
Thread-1 <- "two"
```

masing-masing thread kemudian memanggil method `countSpecialString(string str)` untuk mendapatkan peta karakter beserta jumlah occurencinya 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() {
    }
});
```

Files

- Lengkapilah file [ComplexWordCountServer.java](#)
- Submit kembali file `ComplexWordCountServer.java` yang telah berisi jawaban Anda

Java 8 ▾

 [ComplexWordCountServer.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.14 sec, 31.46 MB

No	Score	Verdict	Description
2	20	Accepted	0.14 sec, 41.29 MB
3	20	Accepted	0.17 sec, 46.41 MB
4	20	Accepted	0.23 sec, 30.26 MB
5	20	Accepted	0.11 sec, 28.50 MB

[◀ Pasca Praktikum 6](#)

Jump to...

⌵

[Praktikum 7 ▶](#)