

[Dashboard](#) / [My courses](#) / [ITB\\_IF2210\\_2\\_1819](#) / [Praktikum 9](#) / [Praktikum 9](#)Question **2**Tries remaining:  
10Marked out of  
80

Time limit	1 s
Memory limit	64 MB

Sebuah Interface **Server** memiliki sebuah fungsi `processRequest` yang menerima sebuah integer request dan menghasilkan sebuah integer lain. Interface **Server** tersebut diimplementasikan oleh 2 buah kelas, yaitu:

**1. AddServer**

Kelas **AddServer** memiliki atribut dan method sebagai berikut

- Atribut private bertipe integer bernama `constant` (didefinisikan sebagai atribut konstan, default = 0)
- Atribut static bertipe integer bernama `numAddServerCreated` yang berisi jumlah **AddServer** yang pernah diinstantiasi
- konstruktor kosong, mengeset nilai default dari atribut `constant`
- konstruktor berparameter yang menerima sebuah integer constant
- Implementasi dari fungsi `processRequest`. Jika nilai `constant`  $\leq 0$ , maka dilakukan penjumlahan kedua angka request. Jika nilai `constant`  $> 0$ , maka dilakukan penjumlahan angka `constant` dengan angka request. Setelah selesai melakukan penjumlahan, cetaklah "Server <numAddServerCreated> execute <angka1> + <angka2>" diakhiri newline dan kembalikan hasil penjumlahan. Contoh (misal `numAddServerCreated` = 1):

1. `constant` = 0 dan request = 5, "Server 1 execute 5 + 5"
2. `constant` = 3 dan request = 5, "Server 1 execute 3 + 5"
3. `constant` = 3 dan request = -5, "Server 1 execute 3 - 5"

**2. MultiplyServer**

Kelas **MultiplyServer** memiliki atribut dan method sebagai berikut

- Atribut private bertipe integer bernama `constant` (didefinisikan sebagai atribut konstan, default = 0)
- Atribut static bertipe integer bernama `numMultiplyServerCreated` yang berisi jumlah **MultiplyServer** yang pernah diinstantiasi
- konstruktor kosong, mengeset nilai default dari atribut `constant`
- konstruktor berparameter yang menerima sebuah integer constant
- Implementasi dari fungsi `processRequest`. Jika nilai `constant`  $\leq 0$ , maka dilakukan perkalian kedua angka request. Jika nilai `constant`  $> 0$ , maka dilakukan perkalian angka `constant` dengan angka request. Setelah selesai melakukan perkalian, cetaklah "Server <numMultiplyServerCreated> execute <angka1> \* <angka2>" diakhiri newline dan kembalikan hasil penjumlahan. Contoh (misal `numMultiplyServerCreated` = 1):

1. `constant` = 0 dan request = 5, "Server 1 execute 5 \* 5"
2. `constant` = 3 dan request = 5, "Server 1 execute 3 \* 5"
3. `constant` = 3 dan request = -5, "Server 1 execute 3 \* -5"

Lalu, buatlah sebuah kelas **Calculator** yang memiliki atribut sebagai berikut

- Atribut private bernama `server` yang mampu menerima kedua jenis server
- Getter atribut `server` (`getServer`)
- Fungsi `addNumber` yang menerima 2 buah angka (`angka1` dan `angka2`) lalu menjumlahkan kedua angka tersebut (`angka1` + `angka2`) dan mencetak "Server returned <hasil>" serta mengembalikan hasil penjumlahan (int).
- Fungsi `doubleNumber` yang menerima 1 buah angka, lalu angka tersebut dikali 2 (`angka` + `angka`) dan mencetak "Server returned <hasil>" serta mengembalikan hasil penjumlahan (int).
- Fungsi `multiplyNumber` yang menerima 2 buah angka (`angka1` dan `angka2`) lalu mengalikan kedua angka tersebut (`angka1` \* `angka2`) dan mencetak "Server returned <hasil>" serta mengembalikan hasil perkalian (int).
- Fungsi `squareNumber` yang menerima 1 buah angka, lalu mengkuadratkan angka tersebut. (`angka` \* `angka`) dan mencetak "Server returned <hasil>" serta mengembalikan hasil perkalian (int).

Kumpulkan **Calculator.java**, **AddServer.java**, **MultiplyServer.java**, **Server.java** dalam **server.zip**

Java 8

