

Started on	Saturday, 3 May 2025, 6:35 PM
State	Finished
Completed on	Saturday, 3 May 2025, 11:11 PM
Time taken	4 hours 36 mins
Grade	300.00 out of 300.00 (100%)

Time limit	1 s
Memory limit	64 MB

Kamu seorang mahasiswa Teknik Elektro yang terkendala untuk melakukan aksi sehari-hari. Akhirnya kamu memutuskan untuk membuat robot sederhana yang bisa membantumu melakukan aksi sehari-hari.

Buatlah sebuah program robot sederhana yang dapat menerima perintah untuk melakukan aksi. Robot dapat menolak perintah apabila perintah yang diberikan tidak sesuai dengan daftar perintah yang dia ketahui.

Lengkapi dan submit file [Robot.java](#).

Sebuah robot memiliki atribut **x** dan **y** yang melambangkan posisi robot. Robot awalnya akan berada pada posisi (0, 0).

Robot bisa melakukan 3 aksi utama, yaitu:

- Walk yang menerima 2 buah parameter, yaitu **x** bertipe **integer** dan **y** bertipe **integer**.
- Talk yang menerima sebuah parameter, yaitu **language** bertipe **string**.
- CheckCommand yang menerima sebuah parameter, yaitu **command** bertipe **string**.


Penjelasan masing-masing method sebagai berikut:

Pada method **walk**, robot menerima parameter **x** dan **y** dan menambahkan nilai itu ke atribut **x** dan **y** yang dimilikinya. Robot lalu akan mengeprint pesan berupa "**Yawes Otw aing (x, y)**" dimana **x** dan **y** adalah nilai akhir setelah penambahan. Apabila nilai **x** atau **y** setelah ditambahkan akan melebihi 5, method ini melemparkan sebuah **exception** dengan pesan "**Malas berjalan**". Nilai **x** dan **y** tidak ditambahkan pada kasus ini.

Robot hanya dapat berbicara menggunakan bahasa Indonesia dan bahasa Inggris saja. Oleh karena itu, pada pemanggilan method **talk**, apabila parameter **language** bukan **Indonesia** atau **Inggris**, robot akan melemparkan sebuah **exception** dengan pesan "**Minimal belajar {language} dulu daripada nyuruh aing doang**". Apabila input valid, robot akan mengeprint pesan berupa "**Jago aing bisa {language}**".

Method **receive command** digunakan untuk mengecek apakah input **command** benar, input **command** yang valid adalah **walk** dan **talk**. Selain kedua command tersebut, method akan melemparkan exception dengan pesan "**Nyuruh-nyuruh aing mulu ih. {command} tidak dikenal**".

Java 8 ▾

 [Robot.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	16	Accepted	0.07 sec, 30.75 MB
2	16	Accepted	0.06 sec, 30.79 MB
3	16	Accepted	0.06 sec, 28.59 MB
4	16	Accepted	0.07 sec, 27.85 MB
5	16	Accepted	0.06 sec, 28.91 MB
6	20	Accepted	0.06 sec, 28.30 MB

Time limit	1 s
Memory limit	64 MB

HTTP Request

Buatlah suatu program yang melakukan pengecekan format untuk HTTP request. Request yang valid harus mematuhi aturan berikut :

- Request tidak boleh kosong ("")
- Request harus memiliki 2 buah slash (/)
- Request hanya bisa mengirimkan HTTP request
- Request hanya bisa mengirimkan metode GET, POST, DELETE dan PUT

Contoh request yang benar : "GET /url HTTP/1.1" => request yang benar (untuk soal ini, asumsikan saja url pasti /url)

contoh request yang tidak valid adalah :

- "" => mengembalikan pesan "Request cannot be empty"
- "GET url HTTP1.1" => mengembalikan pesan "Request has to have 2 /", hal yang sama berlaku untuk request yang punya 1 slash
- "GET /url HTP/1.1" => "Can only send HTTP request"
- "GWT /url HTTP/1.1" => "Method is invalid"

Apabila request mematuhi seluruh aturan, maka request akan mencetak : "Sending <request method> method" request method type disini adalah GET, POST, DELETE atau PUT yang ada pada request

contoh : "GET /url HTTP/1.1" akan mencetak "Sending GET method"

terakhir, apabila request mematuhi seluruh aturan maka ia akan mengembalikan nilai **true**

Lengkapi dan submit file [HTTPRequest.java](#).

File tersebut akan memiliki 3 buah kelas, yaitu **HTTPRequest**, **InvalidRequestException** yang mengextend kelas Exception, dan **InvalidMethodException** yang mengextend kelas Exception. **InvalidRequestException** akan menerima input berupa pesan custom ketika diinisiasi. Sedangkan, **InvalidMethodException** akan mengembalikan string mutlak ketika domain email tidak valid.

Dilarang menambahkan fungsi baru kedalam file **HTTPRequest.java** HINT :

- Untuk mengambil substring dapat menggunakan metode substring!
- Untuk mencari indeks dapat menggunakan metode IndexOf dan lastIndexOf, pelajari penggunaan metode tersebut :D

Java 8 ▾

 [HTTPRequest.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.07 sec, 28.91 MB
2	20	Accepted	0.06 sec, 28.14 MB
3	20	Accepted	0.06 sec, 28.89 MB
4	20	Accepted	0.07 sec, 30.79 MB
5	20	Accepted	0.06 sec, 28.47 MB

Time limit	1 s
Memory limit	64 MB

OwO



Di negara Konoha, semua transaksi hanya bisa dilakukan dengan menggunakan uang tunai. Karena inflasi yang tinggi, semua harga barang menjadi sangat mahal. Akibatnya, masyarakat harus membawa uang dalam jumlah besar ke mana-mana, yang tentu saja tidak praktis.

Kamu, seorang mahasiswa Teknik Informatika, ingin menyelesaikan masalah ini dengan menciptakan aplikasi dompet digital bernama **OwO**. Aplikasi ini ditujukan untuk membantu masyarakat Konoha melakukan transaksi tanpa uang tunai. Namun karena keterbatasan waktu, kamu hanya sempat mengimplementasikan fitur-fitur penting dari aplikasi ini. Namun, kamu tidak ingin aplikasi ini menjadi aplikasi yang buruk. Oleh karena itu, kamu harus memastikan bahwa aplikasi ini memiliki penanganan error yang baik.

Untuk saat ini, kamu hanya diminta untuk mengimplementasikan fitur pengguna OwO di class OwOUser. Aplikasi ini memiliki fitur sebagai berikut:

- Top up saldo. Karena regulasi di Konoha, nominal top up harus lebih dari 1000. Jika kurang, aplikasi akan melempar **InvalidTopUpAmountException** dengan pesan **Top up minimal 1000**.
- Melakukan pembayaran. Nominal pembayaran harus lebih dari 100. Jika kurang, aplikasi akan melempar **InvalidPaymentAmountException** dengan pesan **Pembayaran minimal 100**. Lalu, jika saldo tidak mencukupi, aplikasi akan melempar **InsufficientBalanceException** dengan pesan **Saldo tidak cukup**.
- Cek saldo. Pengguna dapat mengecek saldo saat ini.
- Sebagian fungsi memiliki parameter yang harus valid. Jika tidak, fungsi akan melempar **InvalidParameterException**. Detailnya dapat dilihat di penjelasan fungsi pada file **OwOUser.java** yang sudah disediakan.

Kumpulkan file **OwOUser.java** yang berisi class **OwOUser** dan semua exception yang dibutuhkan. Jangan menambahkan method baru di class OwOUser, dan jangan mengubah nama class yang sudah ada.

Attachments



Attachments: [OwOUser.java](#)

Java 8 ▾

[OwOUser.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.07 sec, 28.86 MB
2	20	Accepted	0.07 sec, 26.35 MB
3	20	Accepted	0.06 sec, 28.23 MB
4	20	Accepted	0.06 sec, 28.63 MB
5	20	Accepted	0.06 sec, 27.97 MB

Jump to...