

Question **1**

Tries remaining:
10

Marked out of
100.00

Time limit	1 s
Memory limit	64 MB

Interface - Vehicle

Anggaplah Anda sedang membuat program sederhana untuk melacak berbagai jenis kendaraan, seperti mobil, sepeda motor, dan sepeda. Setiap jenis kendaraan memiliki karakteristik tertentu

Untuk mengatasi masalah ini menggunakan prinsip OOP, Anda dapat membuat interface untuk semua kendaraan yang disebut "Vehicle". Interface ini akan mendeklarasikan method yang harus dimiliki setiap kendaraan yaitu:

Method Name	Parameter - Type	Return Type
getNumberOfWheels	Empty - Empty	int
getMaxSpeed	Empty - Empty	int
getFuelType	Empty - Empty	String
start	Empty - Empty	void
stop	Empty - Empty	void
turn	direction - String	void

Dengan menggunakan interface seperti ini, Anda dapat memastikan bahwa semua kendaraan memiliki karakteristik dan perilaku yang diperlukan, meskipun implementasi spesifik untuk setiap kendaraan berbeda.

Submitlah file **Vehicle.java** yang merupakan interface berisi method-method pada tabel diatas

Java 8

Maximum size for new files: 512MB, maximum attachments: 1



[Files](#)



You can drag and drop files here to add them.

Run

Check

Question **2**

Tries remaining:
10

Marked out of
100.00

Time limit

1 s

Memory limit

64 MB

Grouped Interface dan Implementasi

Buatlah sebuah class bernama `Point`, yang digunakan sebagai blueprint untuk lokasi. Class `Point` berisi atribut `latitude` dan `longitude`, keduanya memiliki tipe data `double` dan access modifier `private`. Kedua atribut `latitude` dan `longitude` diinisialisai pada konstruktor. Buatlah juga method getter untuk kedua atribut bernama `getLatitude` dan `getLongitude`.

Setelah itu, buatlah sebuah interface lain bernama `Trackable`, yang berfungsi sebagai pemberi informasi tambahan tentang kendaraan, seperti nomor plat dan posisi GPS. Interface ini akan mendeklarasikan beberapa method, diantaranya:

Method Name	Parameter - Type	Return Type
<code>getPlateNumber</code>	Empty - Empty	<code>String</code>
<code>getGPSPosition</code>	Empty - Empty	<code>Point</code>

Terakhir, buatlah interface lain bernama `TrackableVehicle`, yang merupakan Grouped Interface dari interface `Vehicle` dan `Trackable`.

Buatlah `Point.java`, `Trackable.java`, dan `TrackableVehicle.java` tersebut, dan kumpulkan dalam satu ZIP file (**tanpa dibuat folder**) dengan nama bebas. Submit file zip tersebut.

Java 8

Maximum size for new files: 512MB, maximum attachments: 1



[Files](#)



You can drag and drop files here to add them.

Run

Check

Question **3**Tries remaining:
10Marked out of
100.00

Time limit

1 s

Memory limit

64 MB

Interface - Police Car

Buatlah sebuah class bernama **PoliceCar** yang mengimplementasikan interface **TrackableVehicle** yang telah dibuat pada soal nomor 2 yang memiliki atribut sebagai berikut

Nama Atribut	Tipe
plateNumber	String
gpsPosition	Point
speed	Integer
direction	String
wheel	Integer
fuelType	String
started	Boolean

Ketika kelas **PoliceCar** baru dibentuk, atribut **direction** akan secara otomatis bernilai "North" dan **started** akan secara otomatis bernilai false

Kelas akan berisikan **konstruktor** dan **getter** untuk semua atributnya sesuai dengan nama method pada interfacenya. Adapun penjelasan method tambahan sebagai berikut

Metode	Parameter	Return Type	Penjelasan
start	Empty	void	Mengganti started menjadi true
stop	Empty	void	Mengganti started menjadi false
turn	direction	void	Mengganti direction menjadi nilai pada parameter

Submit file bernama **PoliceCar.java**

Java 8

Maximum size for new files: 512MB, maximum attachments: 1

[Files](#)

You can drag and drop files here to add them.

Run

Check

[◀ Feedback Praktikum 3](#)

Jump to...