Question **4**

Tries remaining: 10

Marked out of 100.00

Time limit	1 s
Memory limit	64 MB

Interface - Calculator

Anggaplah Anda sedang membuat program sederhana untuk melacak berbagai jenis kalkulator, seperti scientific calculator, financial calculator, dll. Setiap jenis kalkulator memiliki karakteristik tertentu.

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

Method Name	Parameter - Type	Return Type
add	a,b - int,int	int
substract	a,b - int,int	int
multiply	a,b - int,int	int
divide	a,b - int,int	double
start	Empty - Empty	void
stop	Empty - Empty	void
checkBattery	Empty - Empty	int

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

Submitlah file **Calculator.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 **5**

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 point. Class `Point` berisi atribut `x` dan `y`, keduanya memiliki tipe data `int` dan access modifier `private`. Kedua atribut `x` dan `y` diinisialisai pada konstruktor. Buatlah juga method getter untuk kedua atribut bernama `getX` dan `getY` serta setter `setX` dan `setY`.

Setelah itu, buatlah sebuah interface lain bernama `GraphUpgrade`, yang berfungsi sebagai upgrade calculator dengam method-method untuk kalkulasi Graph, diantaranya:

Method Name	Parameter - Type	Return Type
shiftX	shiftCount - int	Void
shiftY	shiftCount - int	Void
distance	x - int, y - int	double

Terakhir, buatlah interface lain bernama `GraphCalculator`, yang merupakan Grouped Interface dari interface `Calculator` dan `GraphUpgrade`.

Buatlah `Point.java`, `GraphUpgrade.java`, dan `GraphCalculator.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 **6**

Tries remaining: 10

Marked out of 100.00

Time limit	1 s
Memory limit	64 MB

Buatlah sebuah class bernama `CasioGraph` yang mengimplementasikan interface `GraphCalculator` yang telah dibuat pada soal nomor 2 yang memiliki atribut sebagai berikut

Nama Atribut	Tipe
point	Point
batteryLevel	Integer
status	Boolean

Ketika kalkulator hidup, maka status akan bernilai true. Apabila kalkulator tidak hidup, maka status bernilai false.

Ketika kelas `CasioGraph` baru dibentuk, atribut `batteryLevel` akan secara otomatis bernilai 100 dan status yang otomatis bernilai false. Pada konstruktor, juga akan diminta 2 buah parameter x dan y untuk membuat point.

Kelas akan berisikan **konstruktor** dan **getter** untuk semua atributnya sesuai dengan interface yang membangunnya. Adapun penjelasan method tambahan sebagai berikut

Metode	Parameter	Return Type	Penjelasan
add	Int, Int	Int	Menambahkan nilai kedua parameter
substract	Int, Int	Int	Mengurangi nilai parameter pertama sebesar parameter kedua
multiply	Int, Int	Int	Mengalikan nilai kedua parameter
divide	Int, Int	Double	Membagi nilai parameter pertama dengan parameter kedua
start	Empty	void	Menghidupkan kalkulator. Apabila battery = 0, maka kalkulator tidak dapat hidup
stop	Empty	void	Mematikan kalkulator
checkBattery	Empty	Int	Memeriksa battery kalkulator
shiftX	Int	void	Menambahkan nilai X pada point sebesar parameter
shiftY	Int	void	Menambahkan nilai Y pada point sebesar parameter
distance	Int, Int	Double	Menghitung jarak titik pada kelas dengan titik pada parameter dengan rumus V(x1 - x2)^2 + (y1 - y2)^2
getStatus	Empty	Boolean	Mengembalikan status kalkulator
getPoint	Empty	Point	Mengembalikan nilai point yang disimpan oleh kalkulator
charge	Empty	void	Menambahkan batteryLevel hingga 100 dan secara otomatis menyalakan kalkulator

Perlu diperhatikan bahwa setiap menggunakan kalkulator (methods add, substract, multiply, divide, shiftX, shiftY, dan distance) battery kalkulator akan turun sebanyak 10. Apabila kalkulator tidak hidup, maka satu satunya method yang dapat digunakan adalah charge, start, stop, checkBattery, getStatus, dan getPoint. Method lain akan secara otomatis mengembalikan nilai -1 dan metods shiftX dan shiftY tidak akan merubah state kelas.

Submit file bernama `CasioGraph.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...