

Anhaenger

- eigengewicht: int = 1000
- ladung: int = 0
- gebremst: boolean = false

- + Anhaenger()
- + Anhaenger(eigengewicht: int, gebremst: boolean)
- + getter / setter
- + aufladen(zuladung: int): int
- + berechneGesamtgewicht(): int
- + printAnhaenger(): void
- + toString(): String

Wichtig: Der Wert des Attributs ladung wird über die Methode aufladen() verändert. Es wird keine Methode setLadung() benötigt. Den beiden Konstruktor-Methoden wird kein Wert für das Attribut ladung übergeben.

Attribute:

eigengewicht
ladung
gebremst

Hinweise:

- Eigengewicht des Anhängers in kg
- aktuelle Ladung des Anhängers in kg
- der Anhänger ist nicht gebremst (false) oder gebremst (true)

Konstruktoren:

- der paramterlose Konstruktor erzeugt ein Standard-Objekt mit den Default-Werten
- der andere Konstruktor übernimmt beim Instanziiieren das Eigengewicht und den Wert für das Attribut gebremst

aufladen(int: zuladung)

- Wert von zuladung wird zur Ladung addiert
- die Ladung eines gebremsten Anhängers darf maximal das Doppelte seines Eigengewichts betragen
- die Ladung eines ungebremsten Anhänger darf maximal das Einfache des Eigengewichts betragen

Rückgabewert: jene Menge, die nicht mehr aufgeladen werden konnte, weil die maximale Ladung überschritten wurde

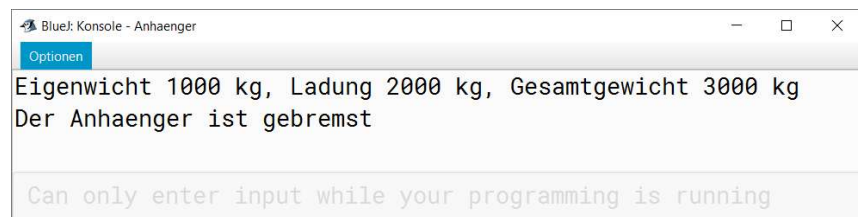
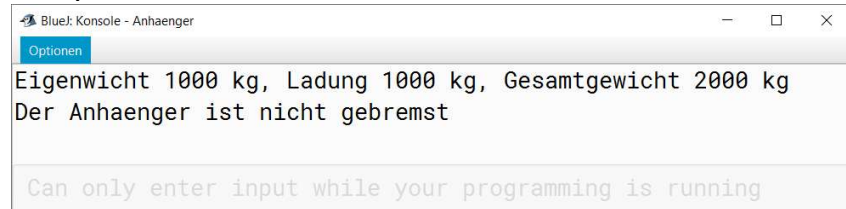
berechneGesamtgewicht():

- gibt das Gesamtgewicht des Anhängers zurück
- das Gesamtgewicht ist die Summe aus Eigengewicht und Zuladung

`printAnhaenger():`

- gibt den Zustand des Anhängers in zwei Zeilen auf der Systemkonsole aus

Beispiele:



`toString(): String`

- liefert den Zustand des Anhängers als String-Referenz zurück, das Layout ist so wie bei den beiden Beispielen bei der Methode `printAnhaenger()` umzusetzen

BONUS-Methode:

`abladen(int): boolean`

- der übergeben kg-Wert wird vom Anhänger “abgeladen”, d.h. der Wert von ladung wird um diesen Wert reduziert und es wird `true` zurückgeliefert. Kann nicht abgeladen werden (weil der Wert von ladung kleiner ist als die “abzuladenden” kg) oder ist der übergebene kg-Wert nicht gültig (weil 0 oder negativ) dann wird `false` zurück geliefert. Bei jedem Fehlerfall wird eine “sprechende” Fehlermeldungen (d.h. ein den Fehler erklärender Text) auf der Systemkonsole ausgegeben.

Testen:

- Erstellen Sie eine Testklasse `TestAnhaenger`
Implementieren Sie zumindest für die Methoden `aufladen(int): int` und `berechneGesamtgewicht(): int` eigene Testmethoden in der Testklasse:

- + `testAufladen(): void`
- + `testBerechneGesamtgewicht(): void`

Viel Spaß und gutes Gelingen!