

Praktikum 3: Vererbung und Polymorphie

Einleitung

Sie programmieren in diesem Praktikum ein kleines Werkzeug zur Berechnung von Kennzahlen von Raketen. Hierzu verwaltet eine Klasse **Rakete** eine Datenstruktur mit (abstrakten) **Bauteilen**.

Diese Bauteile können sein: Container zum Transport von Treibstoff oder Nutzlast, Triebwerke und Sensoren.

Bauteile werden durch eine abstrakte Klasse dargestellt und haben folgende Eigenschaften:

- Eine Bezeichnung mit dem Namen des jeweiligen Bauteils
- Eine Methode zum aktivieren des Bauteils, falls sinnvoll.
- Eine Information, ob das Bauteil aktiv ist.
- Ein Gewicht in kg

Sensoren erben von Bauteil, haben aber keine besonderen Merkmale.

Die **Container** erben von Bauteil und besitzen darüber hinaus folgende Merkmale:

- Maximales Füllvolumen (in kg)
- Füllstand (in **Prozent**)

Beispielsweise soll ein kleiner Treibstofftank ein Leergewicht von 1 kg sowie Platz für 20 kg Treibstoff haben. Ist dieser nun zu 50% gefüllt, so hat dieses Bauteil ein Gesamtgewicht von 11 kg.

Triebwerke haben diese Eigenschaften:

- Schub (in kN, Kilonewton)
- Treibstoffverbrauch (in kg pro Sekunde)

So liefert ein beispielhaftes fiktives Triebwerk den Namen „LV-T30“ einen Schub von 240 kN bei einem Treibstoffverbrauch von 0,5 kg/s.

Sie sollen nun ein textbasiertes **Menü** implementieren, welches den Benutzer aus einer Reihe von vorgegebenen Bauteilen eine **Rakete** zusammenstellen lässt.

Das Menü soll die Möglichkeit bieten, eine Liste aller vorgegeben Teile auf dem Bildschirm auszugeben, Teile zur Rakete hinzuzufügen sowie die Rakete mit allen verbauten Teilen auszugeben. Zusätzlich zur Teileliste sollen bei der **Rakete** folgende Werte berechnet und ausgegeben werden:

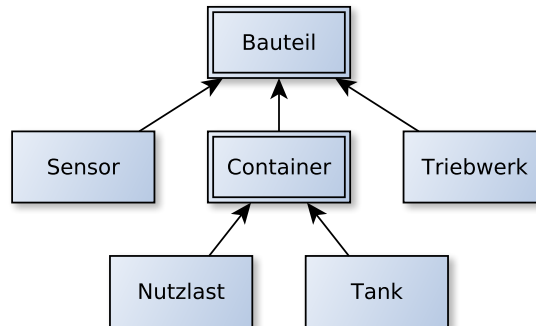
- Gesamtgewicht
- Verhältnis von Schub zu Gewicht (**TWR**¹, thrust-to-weight ratio)
- Brenndauer der Triebwerke

Wichtiger Hinweis: Alle aus der Physik benötigten Formeln sind angegeben, d.h. verfallen Sie nicht in Panik wenn Sie mit den Begriffen „Schub“, „Kilonewton“ oder „TWR“ nichts anfangen können!

¹https://en.wikipedia.org/wiki/Thrust-to-weight_ratio

1 Bauteil-Klassen

Implementieren Sie die Klassen **Bauteil**, Container, Tank, Nutzlast, Triebwerk und Sensor mit folgender Vererbungshierarchie:



Die doppelt umrahmten Klassen sollen **abstrakt** sein!

2 Klasse Rakete

Implementieren Sie eine Klasse **Rakete**, welche eine Datenstruktur mit **Bauteil*** enthält. Dort sollen die Bauteile gespeichert werden, aus denen die Rakete bestehen soll. Außerdem erhält jede Rakete einen Namen.

Die Klasse soll dem Benutzer die Möglichkeit bieten, Bauteile hinzuzufügen und auch wieder zu entfernen. Zum Hinzufügen eines Bauteils sollten Sie eine Methode vorsehen, die das hinzuzufügende Bauteil übergeben bekommt. Die Übergabe soll als **Bauteil*** realisiert werden. Die Klasse **Rakete** erhält die Kontrolle über das Objekt und muss dieses wieder löschen, wenn es nicht mehr benötigt wird.

Weiterhin sollen Methoden existieren, mit denen die folgenden Kennzahlen berechnet werden können:

2.1 Gesamtgewicht

Die Summe aller Gewichte der Bauteile. Bedenken Sie bei der Implementierung von **Container** die Besonderheit des Füllstandes.

2.2 „Thrust-to-Weight Ratio“

Die **TWR** gibt das Verhältnis von Schub zu Gewicht an. Dies gibt u.a. an, ob die Rakete überhaupt genügend Schub besitzt, um starten zu können. Das mittlerweile nicht mehr geflogene Space Shuttle hat beispielsweise einen TWR von 1,5 beim Start. Eine Rakete mit einem TWR unter 1 würde gar nicht erst abheben.

Die Formel hierzu lautet:

$$TWR = \frac{\text{Schub in } N}{(\text{Gewicht in } kg) \times (\text{Erdbeschleunigung})}$$

Hierzu benötigen Sie folgende Werte:

1. Das Gesamtgewicht der Rakete (siehe Unteraufgabe 2.1)
2. Den aufaddierten Schub **aller** verbauten Triebwerke in Newton
3. Die Erdbeschleunigung: $9,807 \text{ m/s}^2$

Hinweis: Der Schub wird in der Regel in kN angegeben, es gilt: $1 \text{ kN} = 1000 \text{ N}$

2.3 Brenndauer

Berechnet wie lange die Triebwerke brennen, wenn **alle** Triebwerke gleichzeitig gezündet würden.

Die Brenndauer berechnet sich aus dem Gesamtverbrauch pro Sekunde aller Triebwerke und der Gesamtheit des Treibstoffes in allen **Treibstofftanks**.

3 Managerklasse

Implementieren Sie eine Managerklasse, die Datenstrukturen für folgende Objekte enthält:

- Raketen
- Triebwerke
- Treibstofftanks
- Nutzlastbehälter
- Sensoren

Es sollen beliebig viele Raketen angelegt werden können.

Die übrigen Datenstrukturen sollen beim Erzeugen der Managerklasse mit den Informationen aus den beiliegenden Textdateien gefüllt werden. Implementieren Sie das Einlesen der Dateien im Konstruktor der Managerklasse.

Implementieren Sie eine Methode `run()`, welche eine Menü ausgibt das folgende Aktionen erlaubt:

- Auflisten aller Raketen (nur die Namen)
- Auflisten aller Triebwerke
- Auflisten aller Treibstofftanks
- Auflisten aller Nutzlastbehälter
- Auflisten aller Sensoren
- Anlegen einer Rakete
- Löschen einer Rakete
- Auswahl einer Rakete

Die folgenden Menüpunkte arbeiten alle auf **einer** Rakete, die im vorigen Menüpunkt ausgewählt wurde.

- Auflisten aller Bauteile
- Hinzufügen eines Bauteils (Fragen Sie danach den Benutzer folgende Informationen:)
 - Art des Bauteils?
 - Name oder Nummer (aus den Auflistungen) des Bauteils?
 - Gegebenenfalls: Füllstand?

Fügen Sie anschließend das gewählte Bauteil der Rakete hinzu. Jedes Bauteil ist standardmäßig inaktiv.

- Löschen eines Bauteils
- Aktivieren aller Bauteile
- Ausgabe der Informationen Gesamtgewicht, TWR und Brenndauer (in einer Ausgabe)

4 Zusatzaufgabe

Erweitern Sie die Bauteile um eine weitere Klasse `Solarpanel`, welches eine gewisse Anzahl an elektrischen Strom erzeugt (in W), falls das Panel aktiv ist.

Fügen Sie weiterhin bei der Klasse `Sensor` einen Stromverbrauch ein.

Bei der Ausgabe aller Informationen einer Rakete geben Sie zusätzlich das Strombudget aus, also ob die verbauten Solarpanels ausreichen, um die Sensoren zu betreiben bzw. wie viel Watt fehlen oder zu viel sind.

In den Materialien finden sie eine aktualisierte Liste der Sensoren, sowie eine Liste der Solarpanels.