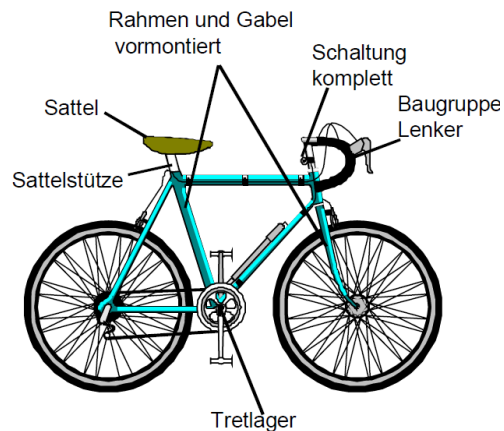


Aufgaben zur Abgabe als Prüfungsleistung für Prog II

- Allgemeine Regeln und Abgabeformat wie in Prog I

Stücklisten

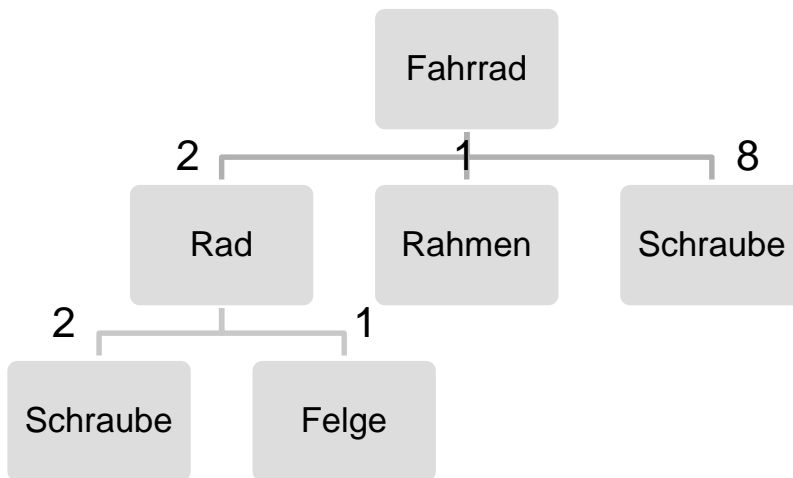
- Stücklisten sind überall dort anzutreffen, wo Endprodukte aus mehreren Ausgangsteilen oder -stoffen zusammengesetzt werden
 - Beispiele: Montage von Geräten, Konfiguration von Fahrzeugen, Rezepturen in der Lebensmittelindustrie
 - Stücklisten bilden in der BWL die Grundlage der Bedarfsermittlung zur Beschaffung bzw. Produktion



Fertigungsstruktur	Produkt X_1
0 (Produkt)	
1 (Baugruppen)	
2 (Einzelteile)	

Anforderungen

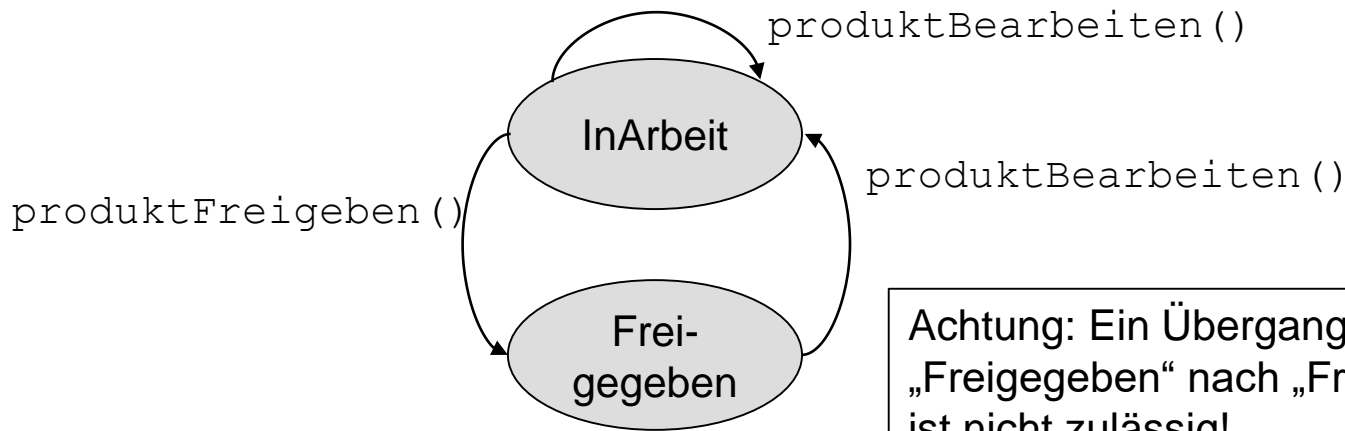
- Die Struktur einer Stückliste soll mittels des Entwurfsmusters „Kompositum“ umgesetzt werden
 - Oberklasse `Produkt` (abstrakte Klasse statt Interface)
 - Abgeleitete Klassen `ProduktStueckliste` bzw. `ProduktBlatt`
- Ein Produkt besitzt unter anderem die Eigenschaften `name (String)`, `anzahl (int)`, `ebene (int, d.h. Ebene 0 für oberste Ebene, Ebene 1 usw.)`
 - Eine `drucken ()`-Methode gibt ein Stücklisten-Objekt mit Einrückung aus:



```
Stueckliste: 1 x Fahrrad
  Stueckliste: 2 x Rad
    ProduktBlatt: 2 x Schraube
    ProduktBlatt: 1 x Felge
  ProduktBlatt: 1 x Rahmen
  ProduktBlatt: 8 x Schraube
```

Anforderungen

- Ein Stücklisten-Objekt besitzt einen Zustand: „InArbeit“ oder „Freigegeben“
 - Der Zustand ist nur für die oberste Ebene von Bedeutung, tiefere Ebene werden ignoriert; der Zustand einer neuen Stückliste ist „InArbeit“
 - Es soll das Entwurfsmuster „Zustand“ verwendet werden mit u.a. Übergangsdiagramm
 - **Klassen:** Zustand, ZustandInArbeit, ZustandFreigegeben
 - **Hinweis:** Man kann das Stücklisten-Objekt im Produkt-Konstruktor mittels `ZustandInArbeit(this)` an die Zustands-Klassen übergeben



Achtung: Ein Übergang von „Freigegeben“ nach „Freigegeben“ ist nicht zulässig!

Anforderungen

- Weitere Methode

- `speichern()` ist für das Speichern einer Stückliste in eine Datei zuständig. Um die Datei aus verschiedenen Klassen beschreiben zu können, ist eine Klasse `SingletonDatei.java` anzulegen, die das Muster Singleton für ein `PrintWriter`-Objekt (siehe Prog I) implementiert.
- Der Pfad der Datei ist hartcodiert `C:\\Java\\stueckliste.txt`
- Die Datei soll eine Textdatei sein und speichert nur die einzelnen Bestandteile der Stückliste mit Einrückungen, aber ohne Klasseninformation (siehe unten)
- Bitte beachten, dass die Datei erst dann geschlossen werden darf, wenn alle Bestandteile geschrieben sind

```
1 x Fahrrad
    2 x Rad
        2 x Schraube
        1 x Felge
    1 x Rahmen
    8 x Schraube
```

Anforderungen

- Weitere Methode
 - In der Klasse `ProduktStueckliste` soll es eine Methode `entfernen(String produktMussWeg)` geben, die mittels übergebenem Produktnamen alles mit diesem Namen aus der Stückliste entfernt, auch wenn das zu entfernende Produkt in einer beliebig tiefen Ebene liegt; falls das Produkt mehrfach in der Stückliste enthalten ist, soll es auch mehrfach entfernt werden
 - Diese Methode ist schwer zu implementieren, da Vorlage aus wikipedia nur direkte Kinder löscht, aber eine Weitergabe an deren Kinder etc. notwendig ist; falls es nicht klappt, bitte Kommentar und weglassen (minus 2 Punkte)
- Hinweis zur Punktevergabe
 - Korrekte Verwendung des Kompositum-Musters: 3 Punkte
 - Korrekte Verwendung des Zustands-Musters: 3 Punkte
 - Korrekte Verwendung des Singleton-Musters: 2 Punkte

Anforderungen

- Im Hauptprogramm `Main.java` sollen folgende Tests implementiert sein:
 1. Aufbau der Stückliste für voriges Fahrrad und Ausgabe der gesamten Stückliste auf der Konsole
 2. Speichern des Fahrrads in der Datei
 3. Test von zweimal Freigeben und zweimal Bearbeiten, erwartete Ausgabe:
 - Neuer Zustand: Freigegeben
 - Funktioniert nicht: Doppelte Freigabe
 - Neuer Zustand: In Arbeit
 - Funktioniert: In Arbeit bleibt
 4. Entfernen der Schrauben und Ausgabe der Stückliste auf der Konsole
- Geben Sie zusätzlich zum Quellcode ein Klassendiagramm (nur Klassen und Attribute, keine Methoden) gemäß Notation der Vorlesung mit ab (mit Kardinalitäten)
 - Format: pdf, auch handschriftlich gezeichnet und dann eingescannt möglich