

Praktische Informatik 1

Objektorientierte Software-Konzepte und UML

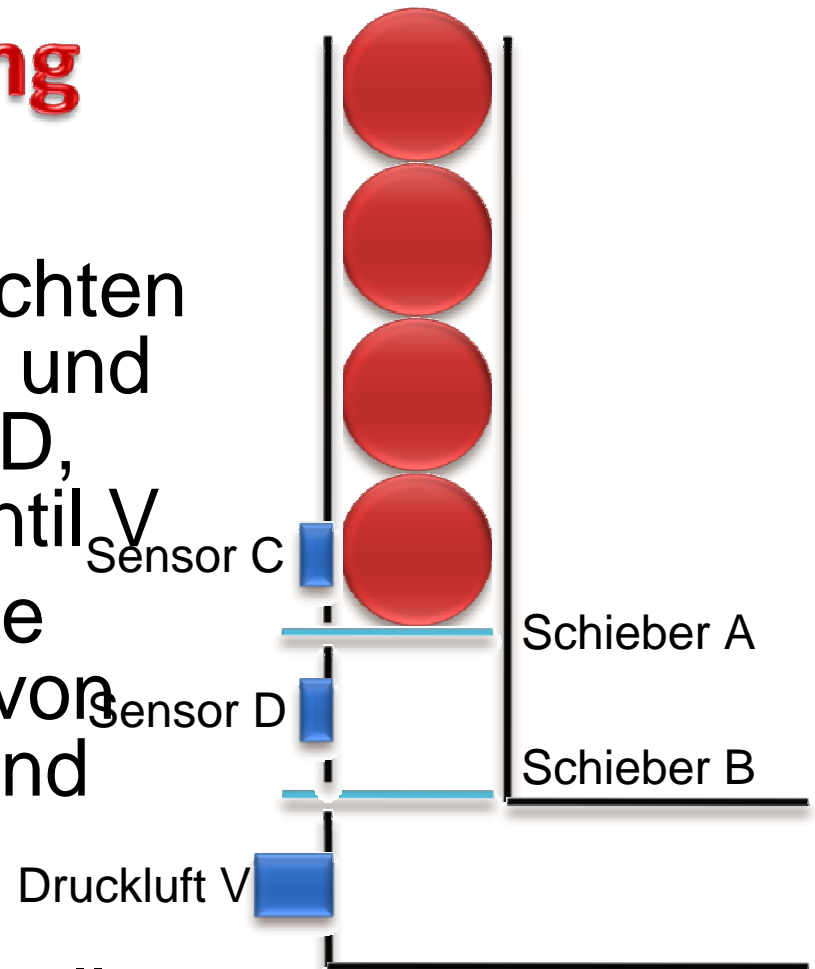
Thomas Röfer

- Funktionale Dekomposition vs. Objektorientierter Ansatz
- Objekte und Klassen
- Objektbeziehungen
- Beispiel
- Musterlösung, Übungsblatt

Objektorient. Analyse und Entwurf

Situationsbeschreibung

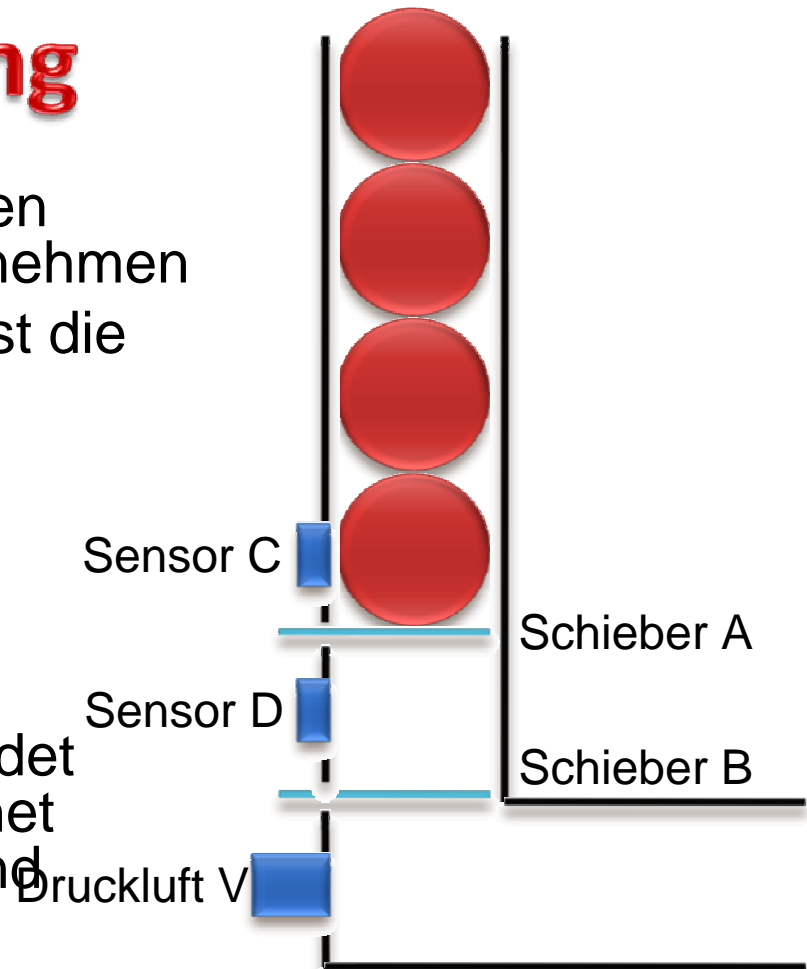
- Die Vereinzelungseinheit besteht aus einer senkrechten Röhre, zwei Schiebern A und B, zwei Sensoren C und D, sowie einem Druckluftventil V
- In der Röhre können Bälle gespeichert werden, die von oben zugeführt werden und die auf Anforderung unten einzeln aus der Röhre geblasen werden sollen



Objektorient. Analyse und Entwurf

Situationsbeschreibung

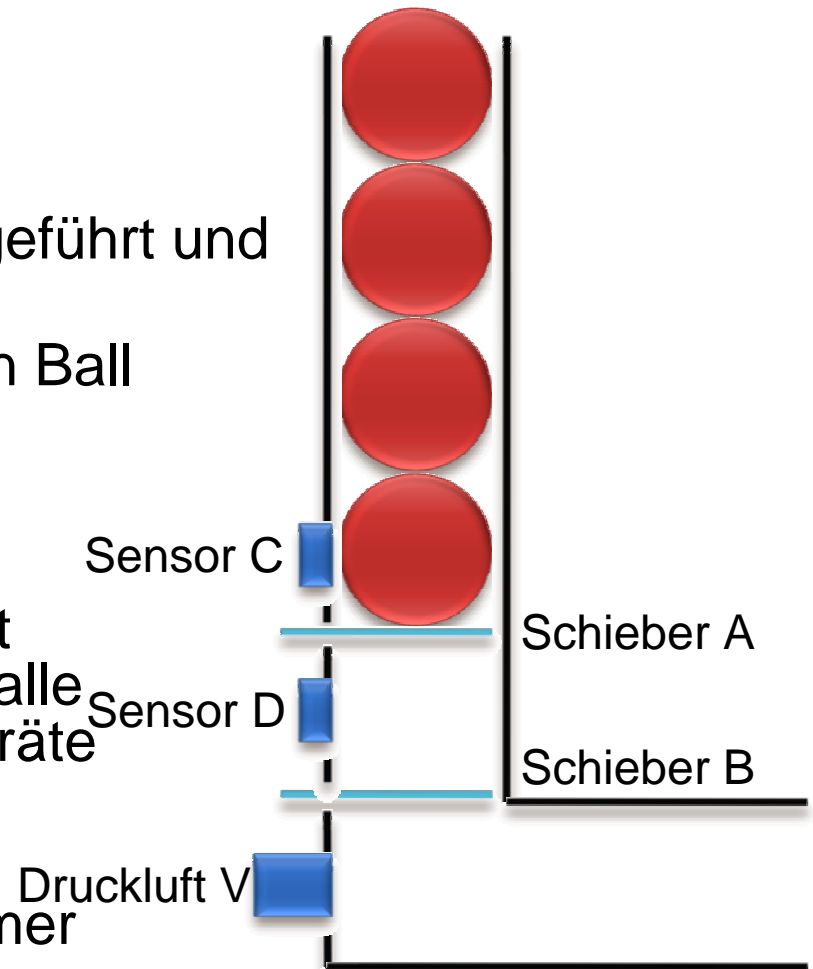
- Die Schieber können die Stellungen „geöffnet“ oder „geschlossen“ einnehmen
- Ist ein Schieber geschlossen, so ist die Röhre geschlossen
- Die Sensoren melden, ob sich an der entsprechenden Stelle ein Ball befindet; falls ja, sind sie aktiviert, falls nein, sind sie deaktiviert
- Am unteren Ende der Röhre befindet sich ein Druckluftventil, das geöffnet oder geschlossen werden kann und das nicht ständig geöffnet bleiben sollte



Objektorient. Analyse und Entwurf

Nutzungsszenarien

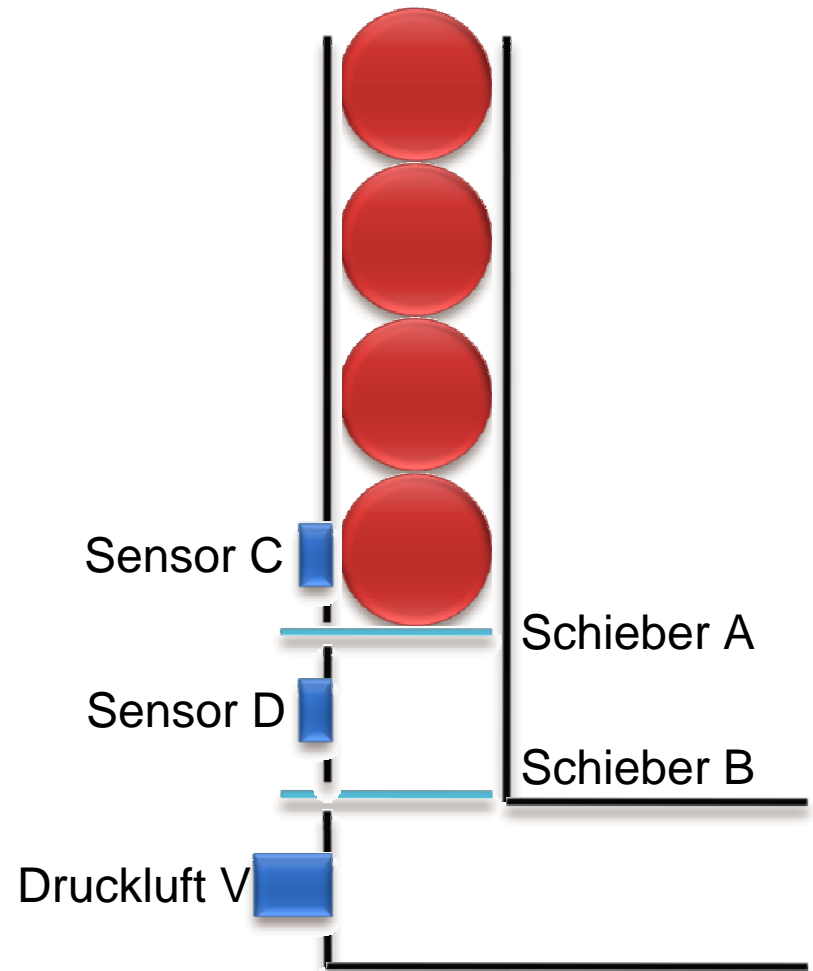
- Standardoperation
 - Die Bälle werden von oben zugeführt und von der Röhre gespeichert
 - Auf Anforderung wird genau ein Ball freigegeben und aus dem Ausgabeschacht geblasen
- Wartung/Fehlerbeseitigung
 - Der Wartungstechniker aktiviert einen Wartungszyklus, in dem alle Operationen der beteiligten Geräte einmal ausgeführt werden
 - Stellt er eine Fehlfunktion fest, so lässt er sich die Seriennummer des betreffenden Gerätes ausgeben und tauscht es aus



Objektorient. Analyse und Entwurf

Mögliche Objekte

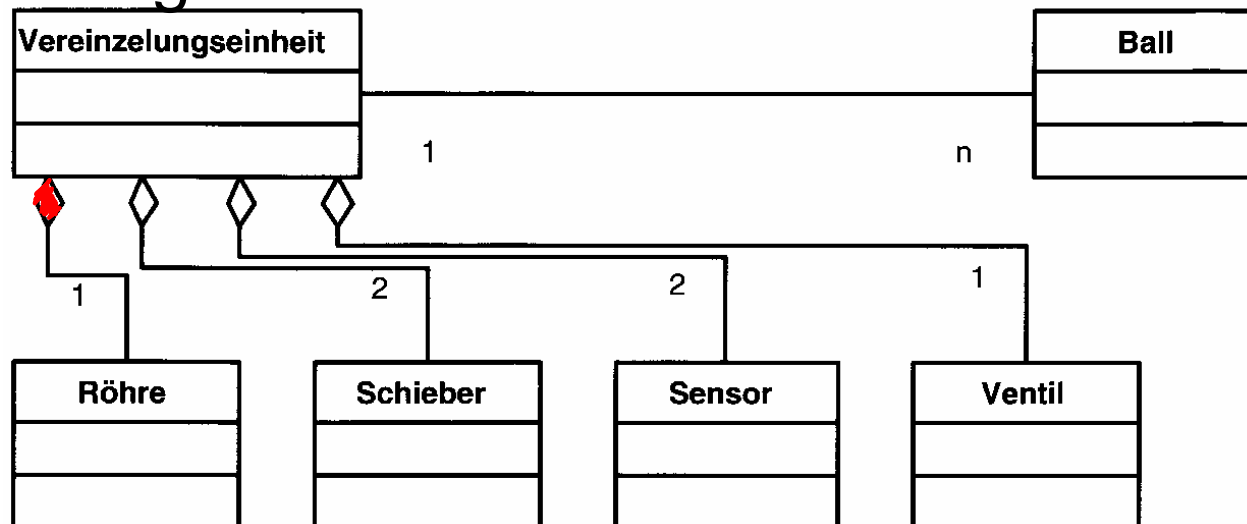
- Vereinzelungseinheit
- Schieber
- Sensor
- Ventil
- Ball (?)
- Röhre (?)



Objektorient. Analyse und Entwurf

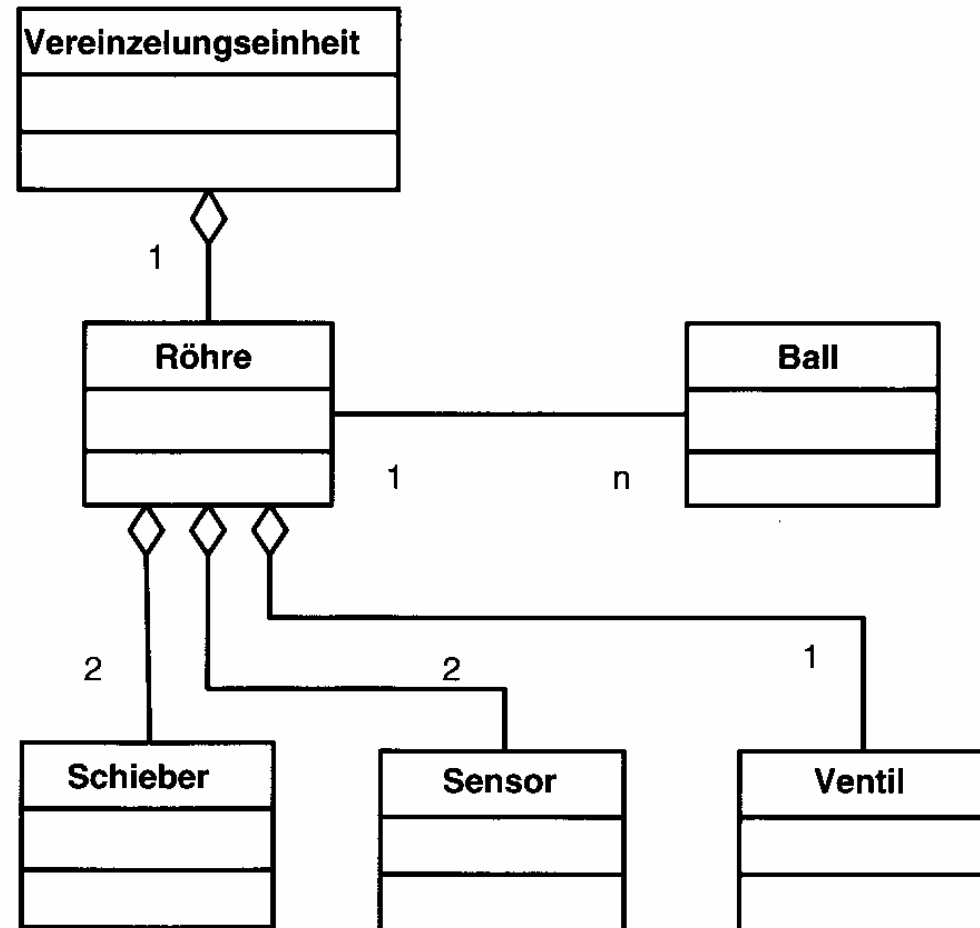
Objekt-Beziehungen

- Die Vereinzelungseinheit enthält Schieber, Sensoren, ein Ventil und eine Röhre
- Bälle sind kein integraler Bestandteil, aber es besteht in jedem Fall eine (eventuell temporäre) Beziehung

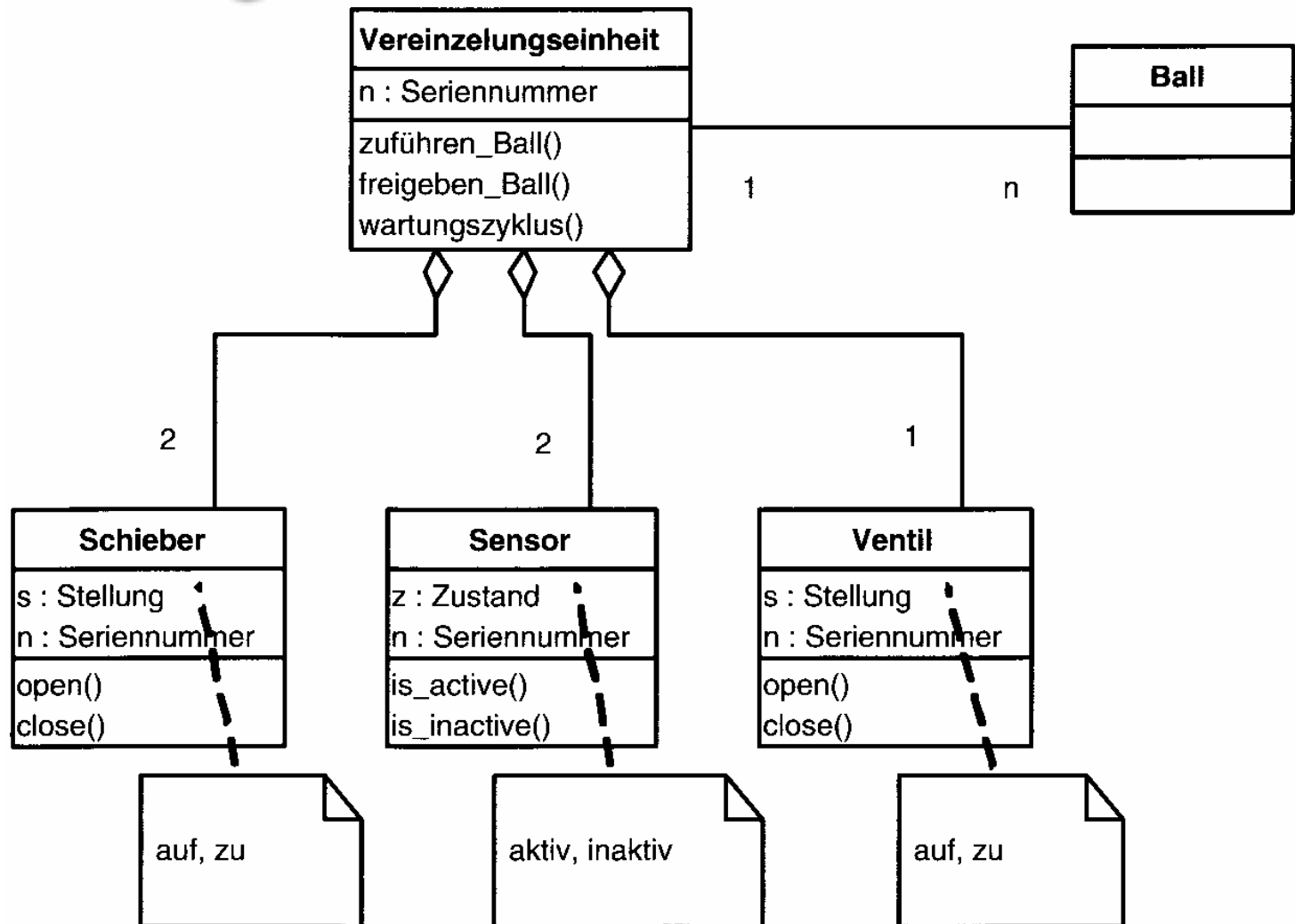


Alternativer Entwurf

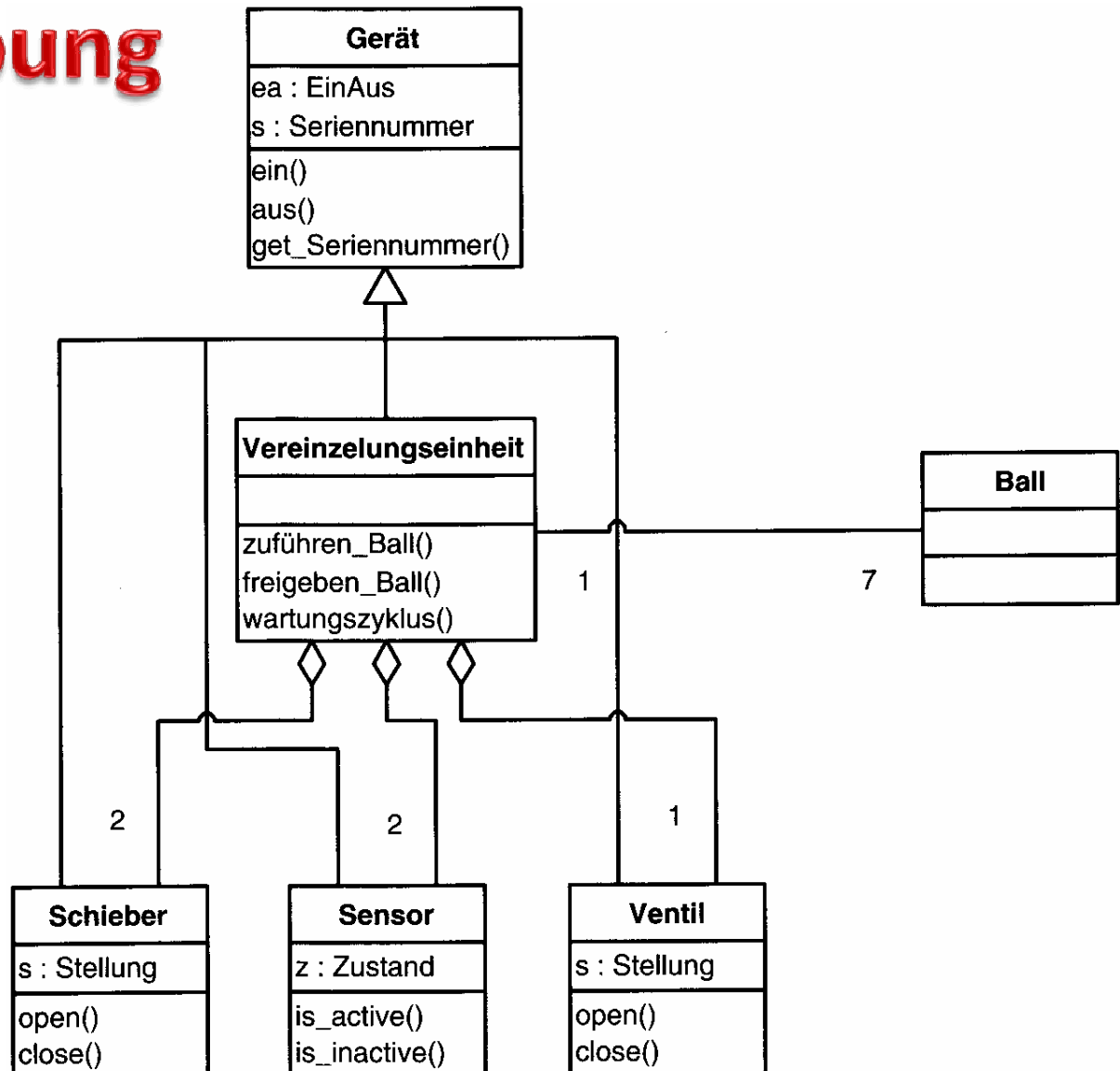
- Aber: Gewöhnliche Röhren enthalten normalerweise keine Schieber und Sensoren
- Daher sollte man eher von einer Vereinzelungsröhre sprechen
- Damit gibt es aber keinen Unterschied zwischen Vereinzelungsröhre und -einheit, denn die Röhre hat keine von der Einheit getrennte Funktionalität



Präzisierung



Mit Vererbung



Musterlösung zu Übungsblatt 2

- Dokumentation soll vor allem beschreiben, warum etwas gemacht wird

- Nicht:
 $i = i + 1; //$ i wird um 1 erhöht

- Testen

- Trivialfälle, Normalfälle, Fehlerfälle, Grenzfälle
 - falls vorhanden
- Tests begründen!

- Vor Abgabe Korrektur

Praktische Informatik I WS 2007/08

Übungsblatt 2 Musterlösung

Aufgabe 1 Bitte ein Bit (30%)

Legt mit BlueJ ein neues Projekt uebung02 an und erzeugt darin eine neue Klasse mit dem Namen Conversion. Fügt zwischen die geschweiften Klammern folgende Methode ein, die Dezimalzahlen in Dualzahlen (als Zeichenkette) wandelt.

1. Beschreibt, was jede einzelne Zeile der Methode tut.

Die erste Zeile enthält die Signatur der Funktion, d.h. sie heißt `dec2bin`, bekommt eine Ganzzahl als Parameter und liefert eine Zeichenkette zurück. Der Parameter hat den Namen `decimal`, enthält also die Dezimalzahl, die in eine Dualzahl gewandelt werden soll.

static zeigt an, dass es sich um eine Klassenmethode handelt, d.h. man kann sie direkt aufrufen, ohne ein Objekt zu instanzieren.

Die Klammern am Ende der ersten und in der neunten Zeile umschließen den Funktionsrumpf.

```
1 static String dec2bin(int decimal) {
```

Die Zeichenkette `digits` speichert die Ziffern, die später verwendet werden, um die Dualzahlen zusammenzusetzen. Da es im Dualsystem nur zwei Ziffern gibt, sind dies nur die 0 und die 1.

```
2     String digits = "01";
```

Die Zeichenkette `binary` wird verwendet, um das Ergebnis aufzubauen. Sie ist anfangs leer und wird später Ziffer für Ziffer erweitert.

```
3     String binary = "";
```

Am Beginn der Schleife wird auf den Trivialfall getestet, d.h. ob die Zahl 0 gewandelt werden soll. In diesem Fall ist das Ergebnis bereits fertig aufgebaut. Ansonsten wird der Schleifenrumpf zwischen den Klammern am Ende von Zeile 4 und in Zeile 7 durchlaufen, solange, bis `decimal` 0 ist.

```
4     while (decimal != 0) {
```

Im Schleifenrumpf wird das Ergebnis aufgebaut, indem immer eine Dualziffer von `decimal` abgespalten wird (durch Modulo 2) und diese vorne an `binary` angefügt wird. `charAt` liefert dabei das n -te Zeichen einer Zeichenkette zurück, in diesem Fall also "0", wenn Divisionsrest 0 ist und "1", wenn er 1 ist.

```
5         binary = digits.charAt(decimal % 2) + binary;
```

Übungsblatt 4

- Aufgabe 1
 - is-a vs. has-a
 - Objektbeziehungen identifizieren
 - Klassendiagramm zeichnen
- Aufgabe 2
 - Code verstehen
 - Fehler korrigieren

Praktische Informatik I WS 2007/08

Übungsblatt 4

Abgabe: 28.11.07

Aufgabe 1 Objektbeziehungen (30%)

- Erklärt den Unterschied zwischen *has-a*- und *is-a*-Beziehung.
- Gibt an, welche Beziehungen (z.B. *has-a* oder *is-a*) zwischen einigen der folgenden Klassen von Objekten bestehen: Büro, Möbel, Tisch, Stuhl, Bein, Schreibtisch, Drehstuhl. Begründet eure Wahl.
- Visualisiert die Klassen und deren Beziehungen in einem Klassendiagramm.

Aufgabe 2 Der TutorInnen Leid... (70%)

TutorInnen haben es nicht leicht. Oft sind Funktionen nicht ganz richtig implementiert und die Fehler müssen gefunden werden. Manchmal gibt es auch keine Dokumentation. Könnt ihr helfen? Ladet dazu das Archiv *uebung04.zip* herunter, packt es aus und nutzt es als Ausgangspunkt für eure Bearbeitung.

Funktion 1. Marla Mystery hat die folgende Funktion abgegeben. Leider ist sie nicht dokumentiert. Was berechnet sie und wie tut sie das?

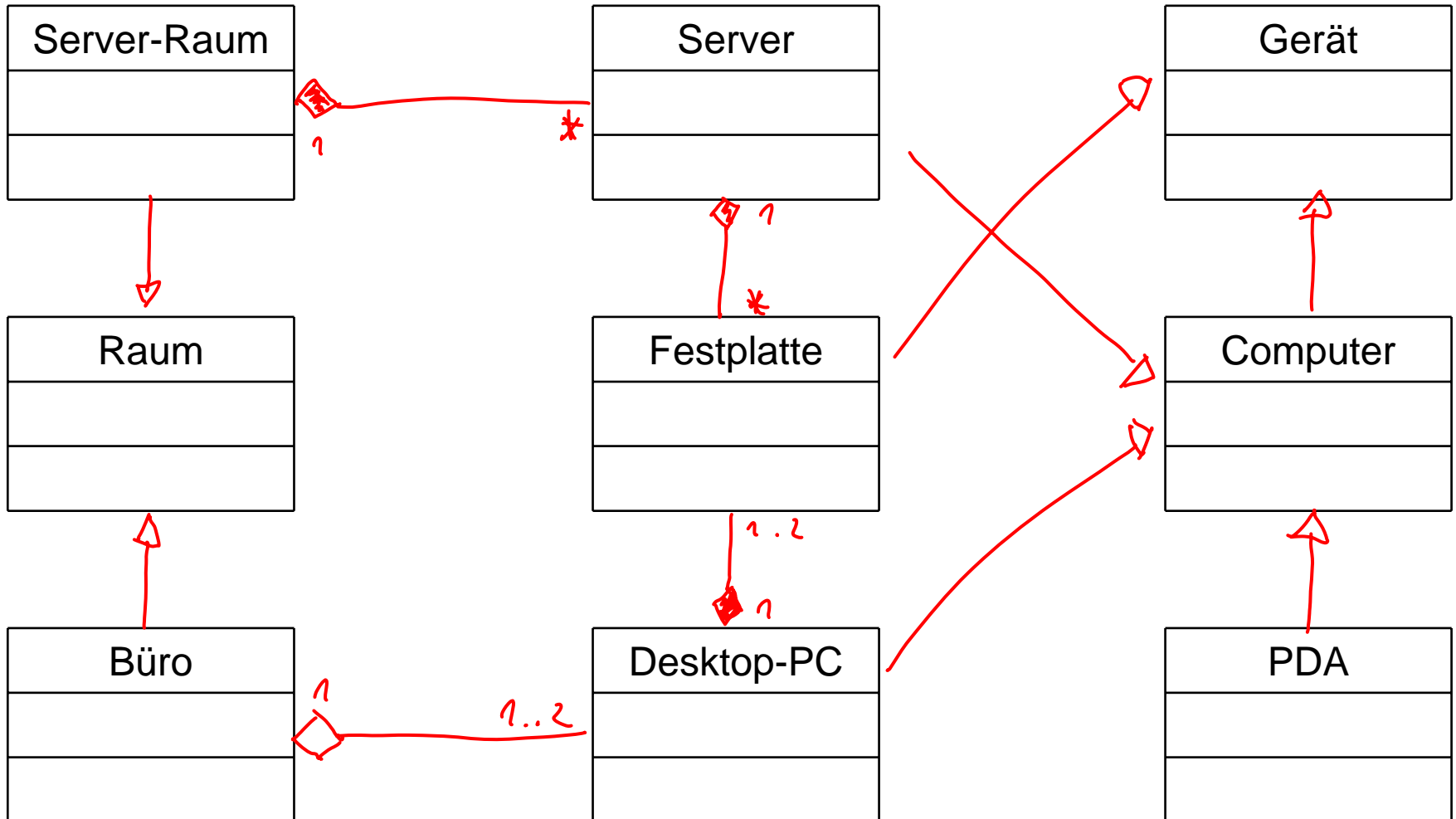
```
1 static int func(int i, int j) {  
2     int r = 1;  
3     int z = j;  
4     while (z > 0) {  
5         r = r * i;  
6         z = z - 1;  
7     }  
8     return r;  
9 }
```

Funktion 2. Man kann in Java auch `$` und `_` zum Benennen von Variablen und Funktionen verwenden. Tut man aber nicht. Karl Krypto hat dennoch die nachfolgende Funktion abgegeben. Was berechnet sie und wie? Tipp: Es hilft nachzuschlagen, was der ternäre Operator `?` macht.

```
1 static int _(int $) {  
2     return $ + $ == $ ? $ + 1 : $ * _($ - $ / $);  
3 }
```

Funktion 3. Karsten Knappdaneben hat die folgende Funktion zur Berechnung des größten gemeinsamen Teilers zweier positiver Zahlen geschrieben. Sie funktioniert auch meistens. Wann nicht? Was muss man ändern?

Übungsblatt 4 – Aufgabe 1



Grafiken in LaTeX einbinden

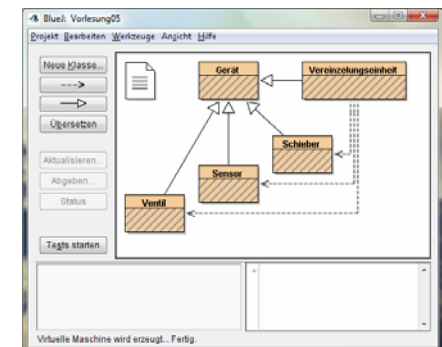
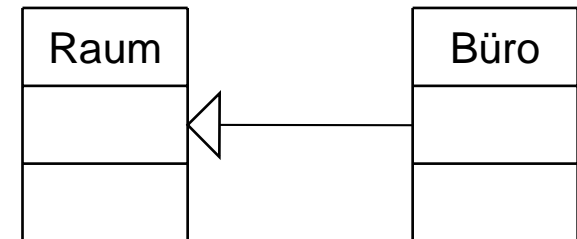
- Paket graphicx.cls
 - Wird in pi1.cls bereits eingebunden

```
\begin{figure}[htb]
  \centerline{
    \includegraphics[width=0.5\textwidth]{diag.pdf}
  }
  \caption{Ein Diagramm\ldots}\label{f:diag}
\end{figure}
```

Wie man in Abb. \ref{f:diag} sehen kann\ldots

Das richtige Grafikformat

- Vektorgrafik
 - Für Zeichnungen
 - Kann verlustfrei skaliert werden
 - **PDF**, SVG, WMF ...
- Rastergrafik
 - Feste Auflösung
 - Verlustfrei komprimiert
 - Für Screenshots
 - **PNG**, BMP, (GIF) ...
 - Verlustbehaftet komprimiert
 - Für Fotos
 - **JPEG/JPG**



Das richtige Grafikformat

- Hallo
- Vektorgrafik
- Rastergrafik
- JPEG

Hallo

Hallo

Hallo

Übungsblatt 4 – Aufgabe 2

•
ode verstehen (Funktionen 1-2)

- Erklären, was die Funktionen tun
- Funktionsweise testen

•
ode korrigieren (Funktionen 3-5)

- Fehler finden
- Nach der Korrektur Funktionsweise durch Testfälle belegen

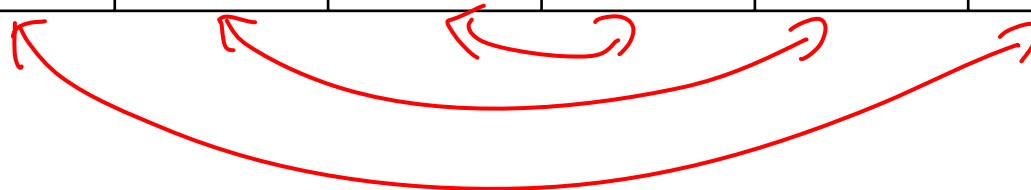
•
esten

- Trivialfälle, Normalfälle, Fehlerfälle, Grenzfälle

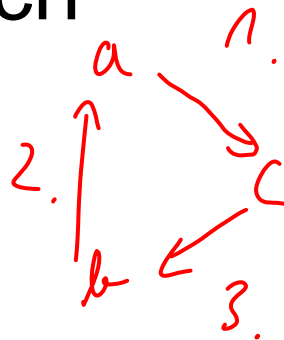
Übungsblatt 4 – Aufgabe 2

Funktion 4

1	2	3	4	5	6
---	---	---	---	---	---



- Dreieckstausch



Übungsblatt 4 – Aufgabe 2

Funktion 5

9	6	7	4	1	3
--------------	--------------	---	--------------	--------------	---

6

7

4

6

7

~~2~~

7

~~6~~

4

8

7

9

7

~~4~~

6

7

4

3

5

7

6

4

Übungsblatt 4 – Aufgabe 2

Funktion 5

1	3	9	7	6	4
---	---	--------------	--------------	--------------	---

		—			
		7	9		
		6		7	
		4	8	7	6
			—		
			7	9	
			6	8	7
7	3	4	6	—	9