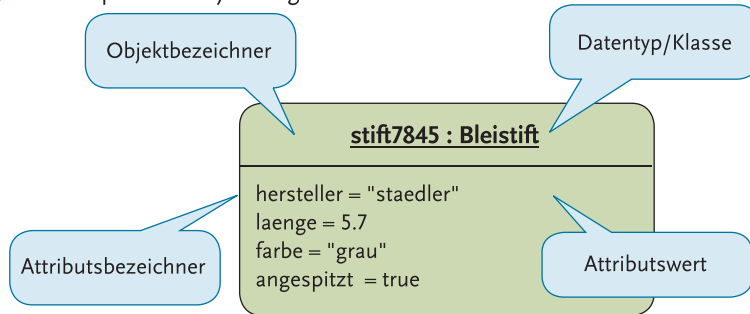


5. Klassentwurf

Anhand einer Schwierigkeitssteigerung sollen die Schülerinnen und Schüler einen einfachen Softwareentwicklungsprozess vom Objekt zum Programm selbstständig durchführen. Ausgehend von realistischen Anforderungen werden über Objekt- und Klassendiagramme Programme in Java implementiert.

5.3 Klassen und Beziehungen entwerfen

1. a)/b) Als Beispiel ein Objektdiagramm:



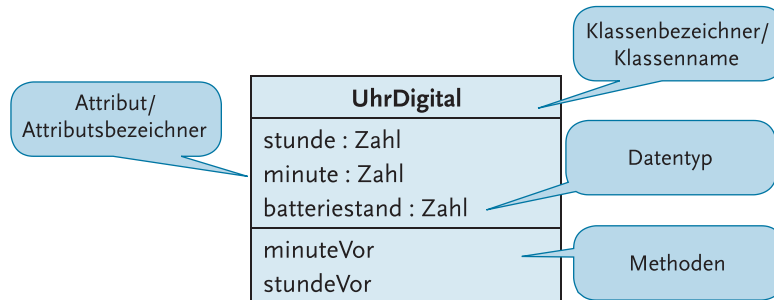
c) Die getroffenen Entscheidungen sollten nachvollziehbar sein. So könnte man alle Arten von Stiften in einer Gruppe (einer späteren Klasse) zusammenfassen, könnte aber auch zwischen Bunt-, Blei- und Farbstiften unterscheiden. Lediglich so genannte Gottklassen wie „Etuiinhalt“ sollten vermieden werden.

d)

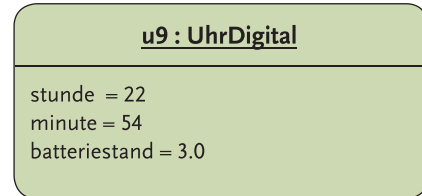
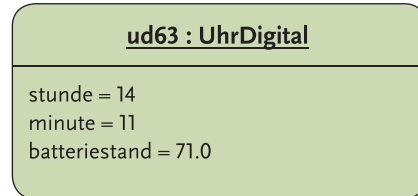
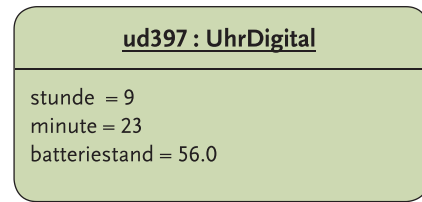
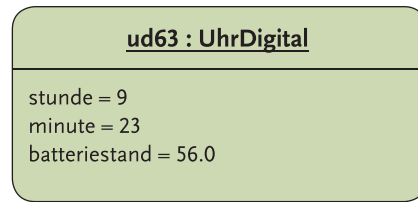
Bleistift
hersteller : Text
laenge : Zahl
farbe : Text
angespitzt : Wahrheitswert
... Methoden ...

e) Die Schülerinnen und Schüler entdecken andere Modellierungsentscheidungen in dem Kurs und setzen sich mit diesen auseinander.

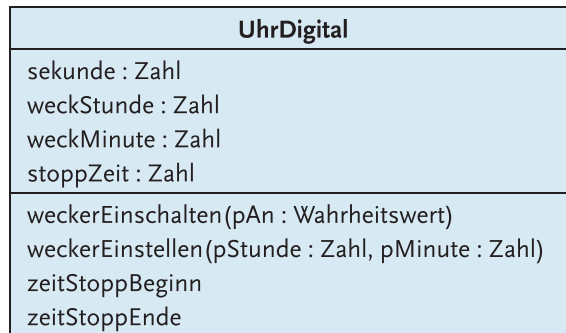
2. a) In einem Objektdiagramm sind die vom Klassendiagramm (Entwurf oder Implementation) festgelegten Attribute mit zum Datentyp passenden Werten belegt. Somit gehört ein Objektdiagramm genau zu einer Klasse, während aus einer Klasse beliebig viele Objekte und somit Objektdiagramme erzeugt werden können.



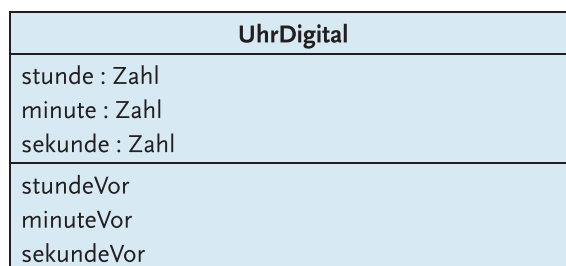
b) Objektdiagramme sind unterschiedlich, wenn sie von einem Objekt unterschiedliche Zustände oder den gleichen Zustand zweier unterschiedlicher Objekte darstellen.



- c) Für diese Aufgabe kann es unterschiedliche Erweiterungen geben. Im folgenden Diagramm gibt es die Möglichkeit, den Wecker ein- oder auszuschalten. Zudem kann eine Weckzeit über Stunden und Minuten eingestellt werden. Die Stoppuhr kann eine Zeit in Sekunden ohne Zwischenzeit stoppen. Sie benötigt neben Stunde und Minute aber noch die Zeiteinheit Sekunde. Es sind lediglich die Erweiterungen angegeben.

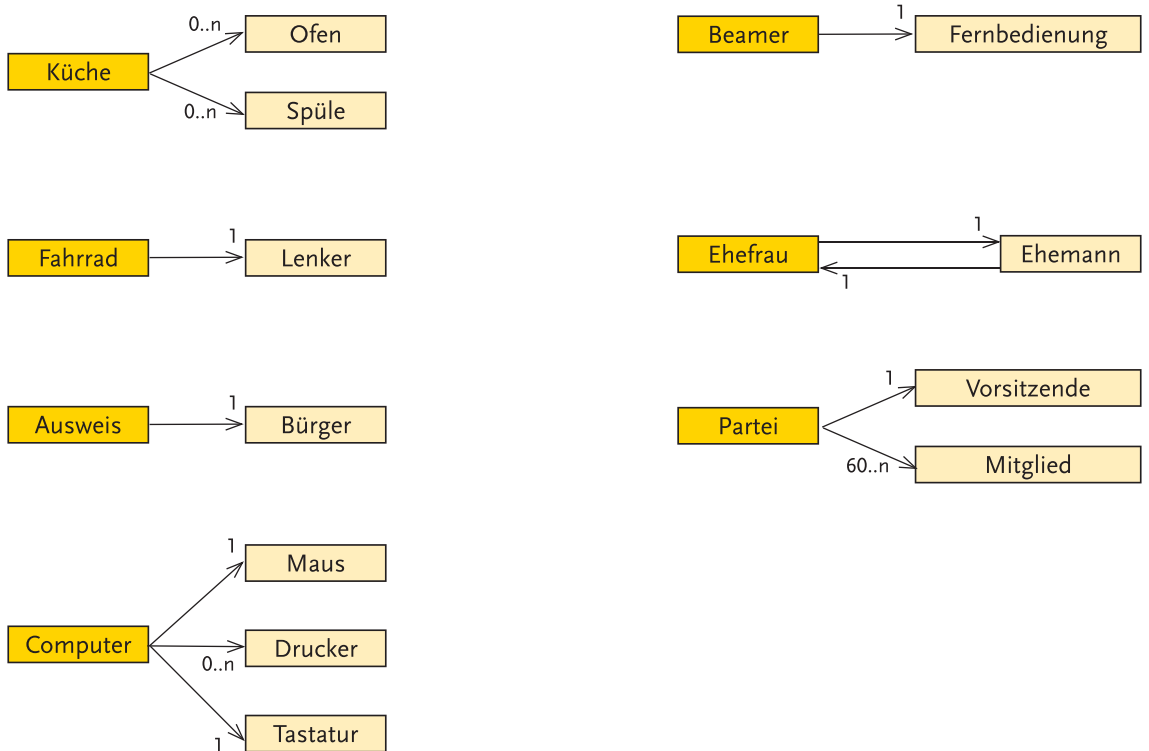


- d) Im Prinzip unterscheiden sich die beiden Entwurfsdiagramme nicht. In einem Objektdiagramm wären 10 Minuten jedoch bei der digitalen Uhr minute=10, bei der analogen Uhr minute = 36 (36°).



SB S. 103

3. Die Lösungen können/sollten Gegenstand von Diskussionen sein, da es einen Interpretationsspielraum gibt.
- Ofen – Spüle – Küche: In einer normalen Haushaltsküche wäre jeweils 1 als Multiplizität denkbar, bei Großküchen nicht.
 - Beamer – Fernbedienung: Eine Fernbedienung wird zwar mit einem Beamer ausgeliefert, könnte jedoch auch andere Beamer verwalten.
 - Lenker – Fahrrad: Ein Lenker pro Fahrrad
 - Ehemann – Ehefrau: Die Multiplizität gilt nur bei Monogamie.
 - Personalausweis – Bürger: Bei mehrfacher Staatsbürgerschaft kann die Multiplizität erhöht werden.
 - Partei – Parteivorsitzender – Parteimitglied: Es werden ungefähr 60 Mitglieder benötigt, um eine Partei zu gründen.
 - Computer – Drucker – Maus – Tastatur: Im Prinzip kann ein Computer weder eine Maus noch eine Tastatur besitzen.



5.4 Klassen und Beziehungen implementieren

SB S. 114

1. a)
- | Zapfsaeule |
|--|
| – preisBenzin : double
– preisDiesel : double
– gewaehlterKraftstoff : String
– abgegebene Liter : double |
| + Zapfsaeule()
+ gibKraftstoffAb(pArt : String, pMenge : double) : void
... alle get- & set-Methoden |
- b)
- | Kamera |
|--|
| – aufloesungHor : int
– aufloesungVer : int
– speicherGesamt : int
– speicherBelegt : int |
| + Kamera(pAH: int, pAV : int, pSG : int, pSB : int)
+ filmen() : void
... alle get- & set-Methoden |
- c)
- | Luftballon |
|--|
| – groesse : int
– farbe : String
– aufgeblasen : boolean |
| + Luftballon(pGroesse: int, pFarbe : String, pAufgeblasen : boolean)
+ luftAufnehmen() : void
+ platzen() : void
... alle get- & set-Methoden |