

# Softwareentwicklung - Aufgaben

## Aufgabe 1

Erläutern Sie den Einsatz von abstrakten Klassen.

(3 Punkte)

## Lösung

In einer abstrakten Klasse werden Gemeinsamkeiten unterschiedlicher Klassen zusammengefasst, sodass die abstrakte Klasse als Basis-Klasse der anderen Klassen eingesetzt werden kann. Typischerweise werden nicht alle Methoden in der abstrakten Klasse implementiert, sondern nur deklariert. Eine Instanzierung abstrakter Klassen ist deshalb nicht möglich.

## Aufgabe 2

bb) Nennen Sie die Aufgabe von Konstruktoren in der objektorientierten Programmierung.

(2 Punkte)

## Lösung

Konstruktoren werden bei der Instanziierung einer Klasse aufgerufen und dienen zur Initialisierung des entstehenden Objektes.

### Aufgabe 3

bc) Nennen Sie die Unterschiede zwischen überschriebenen und überladenen Methoden. (3 Punkte)

### Lösung

Überladene Methoden einer Klasse haben den gleichen Methodennamen aber unterschiedliche Signaturen.

Eine Methode einer Basisklasse wird in einer abgeleiteten Klasse überschrieben wenn diese den gleichen Namen und die gleiche Signatur besitzt.

## Aufgabe 4

bd) Erläutern Sie den Begriff "statische Eigenschaften".

(2 Punkte)

## Lösung

Statische Eigenschaften existieren unabhängig von der Instanzierung der Klasse und sind deshalb nur einmal vorhanden.

## Aufgabe 5

Nennen Sie 2 Vorteile, die ein CASE-Tool bei der Softwareentwicklung bietet.

## Lösung

- Schnellere Umsetzung von Projekten durch:
- Teamarbeit
- Versionsverwaltung
- U. ä.

## Aufgabe 6

Nennen Sie 2 Werkzeuge eines CASE-Tools.

## Lösung

- Debugger
- Diagrammerstellung
- u. ä.

## Aufgabe 7

- d) Sie sollen ein Konzept entwickeln, mit dem bereits während der Softwareentwicklung eine hohe Qualität der Software sichergestellt werden kann. Erläutern Sie drei Maßnahmen, die dieses Qualitätssicherungskonzept enthalten sollte. (6 Punkte)

## Lösung

- enge Einbindung der Benutzer
- ständige UATs (User Acceptance Test) der jeweiligen Prototypen
- kurze Entwicklungsstufen und -schritte
- u. a.

Alternative Lösung möglich

## Aufgabe 8

Für die neue Software wollen die Entwickler der Soft-EDV GmbH „wiederverwendbare Softwarekomponenten“ erstellen.

Nennen Sie je ein Beispiel für wiederverwendbare Softwarekomponenten aus der

- strukturierten Programmierung
- objektorientierten Programmierung.

(4 Punkte)

---

## Lösung

strukturierte Programmierung: externe Unterprogramme

objektorientierte Programmierung: in Java eingebundene Klassen.

## Aufgabe 9

Ein Kollege schlägt vor, anstatt der Observer-Musters zur Aktualisierung des Views Datenbindung (Data Binding) einzusetzen.

c) Erläutern Sie den Begriff.

2 Punkte

---

---

---

---

## Lösung

Als Datenbindung (engl. Data Binding) bezeichnet man z. B. die automatische Weitergabe von Daten von einem Datenobjekt an ein Steuerelementobjekt (Viewobjekt) einer Benutzeroberfläche und umgekehrt.

## Aufgabe 10

Beschreiben Sie eine Möglichkeit, wie in einem Programm festgestellt werden kann, ob der Quotient einer Division ganzzahlig ist.

---

---

---

---

---

---

---

---

---

---

+

## Lösung

Modulofunktion verwenden, die den auf eine ganze Zahl gerundeten Rest eines Quotienten zurückgibt.  
Ist der Rückgabewert 0, ist der Quotient eine ganze Zahl.

oder

Division zwei Mal durchführen.

Einen Quotienten runden.

Gerundeten und ungerundeten Quotienten miteinander vergleichen; sind beide Quotienten gleich, ist der Quotient eine ganze Zahl.

oder

eine Funktion aufrufen

## Aufgabe 11

Erläutern Sie die folgenden vier Schritte bei der Entwicklung von Programmen mit paralleler Datenverarbeitung. 8 Punkte

1. Partitionierung:
2. Kommunikation:
3. Zusammenfassung:
4. Zuordnung:

---

1. Partitionierung:

---

---

2. Kommunikation:

---

---

3. Zusammenfassung:

---

---

4. Zuordnung:

---

## Lösung

1. Partitionierung:

Zerlegung der Gesamtaufgabe in möglichst viele kleine Teilaufgaben

2. Kommunikation:

Ermittlung der Daten, die als Ergebnisse der Teilaufgaben anfallen und Festlegung des Datenflusses zwischen den Teilaufgaben

3. Zusammenfassung:

Bündelung/Zusammenlegung von kleinen Teilaufgaben

4. Zuordnung:

Festlegung der Ausführungsorte (Kerne) für die Teilaufgaben

## Aufgabe 12

ac) Die Öko-Soft GmbH prüft, ob dieses System als klassisches Client-Server-System oder als Webapplikation verwirklicht werden sollte.

Erläutern Sie beide Systeme vergleichend.

(8 Punkte)

## Lösung

8 Punkte

### Client-Server-Lösung

- Geringerer Datentransfer bei der Client-Applikation, da nur die Daten übertragen werden
- Client-Applikation läuft schneller
- Download und Installation der Client-Applikation erforderlich
- Bessere Kontrolle der Nutzer durch gezielte Verteilung der Software
- Evtl. Upgrade erforderlich
- Client-Applikation muss für verschiedene Systeme bereitgestellt werden.
- Versionsunterschiede können zu Problemen führen
- Wartungs- und daher kostenintensiver
- u. a.

### Bei Web-Applikation

- Nur Browser erforderlich
- Auf allen PCs mit entsprechendem Internet-Browser zugänglich
- Direkter Zugang auch für Neukunden, da keine Installation erforderlich
- Darstellungsunterschiede bei verschiedenen Browsertypen möglich
- Ggf. gelockerte Sicherheitseinstellungen beim Browser nötig (wenn mit Cookies gearbeitet wird)
- u. a.

## Aufgabe 13

- db) Die RADL-BLITZ GmbH erhält auch Aufträge von ausländischen Kunden. Erläutern Sie kurz, wie dies bei der Gestaltung des Web-Auftritts zu berücksichtigen ist. (2 Punkte)

### Lösung

- Mehrsprachige Webpräsenz
- Farbliche Gestaltung (unterschiedliche Bedeutung der Farben!)
- Einbindung weiterer Zeichensätze (arabische Schriftzeichen!)

Alternative, in sich schlüssige Lösungen sind auch als richtig zu bewerten.

## Aufgabe 14

Nennen Sie für die serverseitige und die clientseitige Ausführung eines Skriptes

ca) zwei konzeptionelle Gemeinsamkeiten. (2 Punkte)

cb) zwei konzeptionelle Unterschiede. (2 Punkte)

## Lösung

ca) 2 Punkte

- Müssen in die Webseite eingebunden sein
- Schwache Typisierung
- OOP-Strukturen
- u. a.

cb) 2 Punkte

- Unterschiedliche Ausführungsorte der aktiven Programmbestandteile
- Unterschiedliche Anforderungen an Anwendersystem
- Sichtbarkeit des Quellcodes für Anwender
- u. a.

## Aufgabe 15

Beschreiben Sie stichwortartig die Funktion und Arbeitsweise von Assembler, Compiler und Interpreter.

## Lösung

<b>Typ</b>	<b>Beschreibung</b>
Assembler	maschinennah, mnemotechnisch aufgebaut
Compiler	werden am Stück übersetzt, selbstständig lauffähige Programme (bezogen auf ein Betriebssystem)
Interpreter	Skriptsprachen, werden zur Laufzeit übersetzt, benötigen zur Ausführung ein Shell- oder Wirtprogramm

## Aufgabe 16

Tragen Sie die Kennziffern der folgenden Sprachen, die entweder Compiler oder Interpreter zugeordnet werden können, in die untenstehende Tabelle ein. (3 Punkte)

### Sprachen

Nr.	Sprache
1	C
2	C++
3	COBOL
4	HTML
5	Java
6	JavaScript
7	Pascal
8	Perl
9	Smalltalk
10	Visual Basic for Application
11	XML

Übersetzungsprogramm/ Programmierwerkzeug	Sprachen
Compiler	
Interpreter	

Nennen Sie die Sprachen, die der obigen Tabelle nicht zugeordnet werden können, und begründen Sie, weshalb eine eindeutige Zuordnung problematisch ist.

## Lösung

<b>Übersetzungsprogramm/Programmierwerkzeug</b>	<b>Sprachen</b>
Compiler	1, 2, 3, 7, 9
Interpreter	6, 8, 10

5: Sprache wird zwar übersetzt, nicht aber gebunden; zudem wird eine „virtuelle Maschine“ zur Ausführung benötigt

4, 11: Seitenbeschreibungs- bzw. Strukturbeschreibungssprache

## Aufgabe 17

Die Erstellung eines C-Programms erfolgt in mehreren Schritten, auch wenn diese in einer integrierten Entwicklungsumgebung häufig zusammengefasst sind.

Nennen Sie zu jedem Vorgang das benötigte Werkzeug.

(4 Punkte)

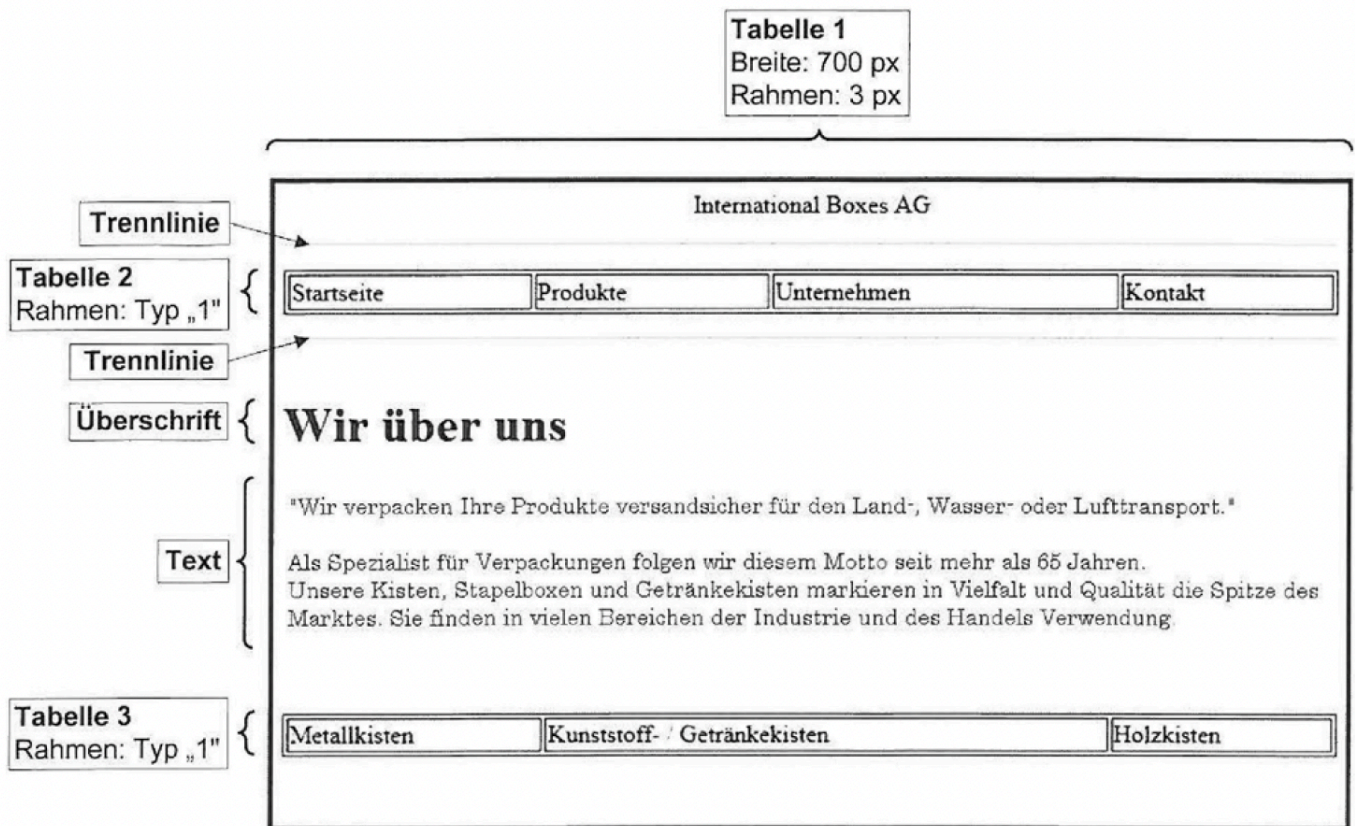
Vorgang	Werkzeug (Fachbegriff)
Quelltext erstellen	
Quelltext übersetzen	
Lauffähigkeit herstellen	
semantische Fehler suchen	

## Lösung

Vorgang	Werkzeug (Fachausdruck)
Quelltext erstellen	Editor
Quelltext übersetzen	Compiler
Lauffähigkeit herstellen	Linker
semantische Fehler finden	Debugger

## Aufgabe 18

Im Zuge der Restrukturierung wird auch der Webauftritt der international Boxes AG überarbeitet. Der erste Entwurf der HTML-Startseite sieht wie folgt aus:



a) Die Struktur der Seite soll durch die Verwendung von HTML-Tabellen realisiert werden.

Vervollständigen Sie den gegebenen HTML-Code derart, dass die Webseite diesen Aufbau besitzt. (13 Punkte)

Hinweis:

Der Text im Zentrum von „Wir verpacken ....“ bis „und des Handels Verwendung.“ muss nicht abgeschrieben werden.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 STRICT/EN"
„http://www.w3.org/TR/xhtml1/DTD/ xhtml1-strict.dtd">
```

```
<html>
```

```
  <head>
```

```
    <title>Willkommen bei der International Boxes AG</title>
```

```
  </head>
```

```
    <body>
```

```
      </body>
```

```
</html>
```

# Lösung

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 STRICT//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
  <title>Willkommen bei der International Boxes AG.</title>
</head>
<body>

<table align="center" cellspacing="5" style="width: 700px;border:3px solid;" >
<tr>
  <td style="text-align:center;">International Boxes AG</td>
</tr>
<tr>
  <td><hr /></td>
</tr>
<tr>
  <td>
    <table border="1" style="width: 100%;">
      <tr>
        <td>Startseite</td>
        <td>Produkte</td>
        <td>Unternehmen</td>
        <td>Kontakt</td>
      </tr>
    </table>
  </td>
</tr>
<tr>
  <td><hr /></td>
</tr>
<tr>
  <td style="padding-bottom: 40px;">
    <h1>Wir &uuml;ber uns</h1>
    <p>Text, der nicht abgeschrieben werden muss </p>
    <table>
      <tr>
        <table border="1" style="width: 100%;">
          <tr>
            <td>Metallkisten</td>
            <td>Kunststoff- / Getr&auml;nkekisten</td>
            <td>Holzkisten</td>
          </tr>
        </table>
      </tr>
    </table>
  </td>
</tr>
</table>
</body>
</html>
```

## Aufgabe 19

b) Die Leitung der International Boxes AG legt besonderen Wert auf eine einheitliche Gestaltung aller Webseiten. Im Head-Bereich der Webseite findet sich folgendes HTML-Tag.

```
<head>
```

```
  <link rel="stylesheet" type="text/css" href="style.css">
```

```
</head>
```

ba) Erläutern Sie die Attribute des HTML-Tags „link“, Welche Auswirkung hat dies auf die Darstellung im Browser? (4 Punkte)

## Lösung

Einbinden des Verweises auf die Datei style.css. Die Datei beinhaltet die Formatierungsinhalte zu den verwendeten Tags.

## Aufgabe 20

- bb) Erläutern Sie, warum die gewählte Realisierung für eine einheitliche Webseitengestaltung (Corporate Design) besonders günstig ist. (4 Punkte)

## Lösung

Datei ist auch für andere HTML-Seiten zu verwenden (CD)

Änderungen des Designs (Schriftart, -größe, Hintergrundfarben, ...) erfolgen nur einmal zentral für alle Dokumente.  
Webseiten können leichter von verschiedenen Personen erstellt werden (Erleichterung des CD).

## Aufgabe 21

bc) Die Webseite erhält nach der Umsetzung folgendes Aussehen.



Durch Mausklick auf eines der gezeigten Produktbilder soll auf die entsprechende Produktwebseite „produktname.html“ verlinkt werden.

Bei der Codierung des HTML-Tags ist Folgendes zu beachten:

- Startseite und Produktwebseite liegen im selben Ordner.
- Alle Produktbilder (produktname.jpg) liegen in dem Unterordner „images“ dieses Ordners.
- Bei nicht verfügbarem Produktbild soll ein Alternativtext angezeigt werden.

Erstellen Sie für das Produkt „Metallkiste“ den entsprechenden HTML-Tag. (4 Punkte)

## Lösung

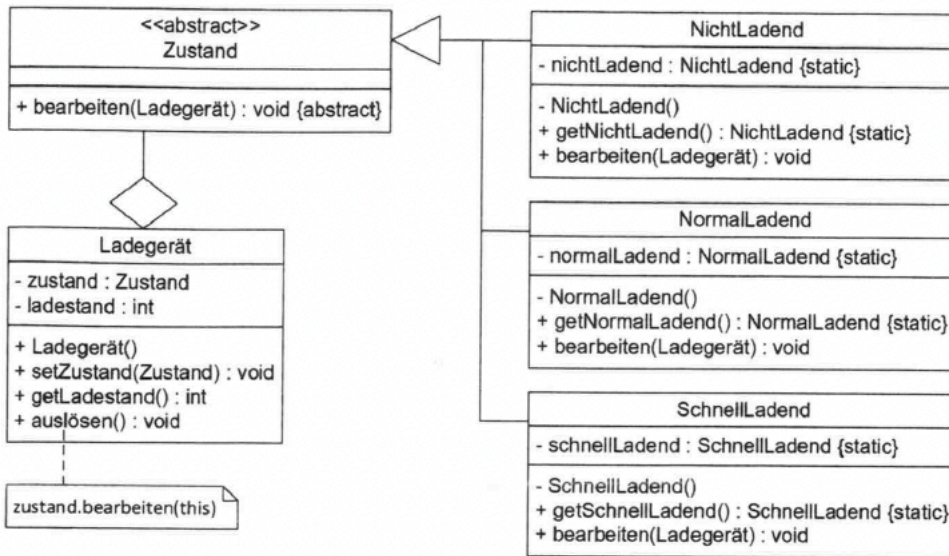
c) 4 Punkte

```
<a href="Metall.html">  
  
Metallkisten  
</a>
```

Andere Lösungen möglich

## Aufgabe 22

b) Die Software soll objektorientiert nach dem Zustand-Entwurfsmuster (State Pattern) programmiert werden. Für die Umsetzung liegt das folgende UML-Klassendiagramm vor.



Hinweis: Notation für UML-Klassendiagramm siehe Belegsatz, Seite 3

ba) Im Konstruktor der Klasse `Ladegerät` wird der Anfangszustand durch Initialisierung von `zustand` mit einem `NichtLadend`-Objekt festgelegt.

Formulieren Sie die entsprechende Anweisung.

2 Punkte

+ `Ladegerät()`

---



---



---

bb) In der `bearbeiten`-Methode der Klasse `NichtLadend` wird bei einem Akku-`ladestand` größer gleich 20 und kleiner 100 der Referenz `zustand` des Ladegeräts ein `NormalLadend`-Objekt zugewiesen.

Formulieren Sie die Kontrollstruktur mit entsprechender Anweisung.

3 Punkte

+ `bearbeiten(ladegerät : Ladegerät) : void`

---



---



---

bc) Erläutern Sie anhand des gegebenen Entwurfsmusters den Begriff Polymorphie. Nutzen Sie dazu die Instanzvariable `zustand`.

4 Punkte

## Lösung

ba) 2 Punkte

1 Punkt für „setZustand“ oder „zustand:=“, 1 Punkt Aufruf des Objektgetters

+ Ladegerät()

entweder über den Setter:

```
setZustand(NichtLadend.getNichtLadend())
```

oder direkt:

```
zustand:= NichtLadend.getNichtLadend()
```

bb) 3 Punkte

1 Punkt Bedingung, 2 Punkte Methodenaufrufe

+ bearbeiten(ladegerät : Ladegerät) : void

wenn (ladegerät.getLadestand() >= 20 und ladegerät.getLadestand()<100)

dann ladegerät.setZustand(NormalLadend.getNormalLadend())

bc) 4 Punkte

Die Klasse Ladegerät hält die polymorphe Referenz *zustand* auf die abstrakte Klasse *Zustand* (1 Punkt). Dieser Referenz werden zur Laufzeit konkrete Objekte der Klassen NichtLadend, NormalLadend bzw. SchnellLadend zugewiesen (1 Punkt). Das ist möglich, da die konkreten Zustandsklassen von der abstrakten Klasse *Zustand* abgeleitet sind (1 Punkt). Je nachdem, welchen konkreten Zustand *zustand* gerade referenziert, wird mit *zustand.bearbeiten(this)* immer das entsprechende zustandsspezifische Verhalten ausgeführt (spätes/dynamisches Binden) (1 Punkt).

## Aufgabe 23

Die bildgebende Diagnostik liefert täglich viele Dateien, die gespeichert werden müssen. Um Speicherplatz einzusparen, soll ein Komprimierungsalgorithmus entwickelt werden. Für einen ersten Prototypen wurde folgende Vorgabe erstellt.

Vorgabe:

Die Bilddaten sollen mit einer Lauflängenkodierung komprimiert werden. Dabei werden sich direkt wiederholende Zeichen zusammengefasst, und nur die Anzahl und das entsprechende Zeichen erfasst. Eine Zusammenfassung soll erst bei mehr als vier Zeichen erfolgen. Zur Erkennung der Lauflängenkodierung wird das „%“-Zeichen verwendet, das in den unkomprimierten Bilddaten nicht vorkommt. Die Bilddaten liegen als String-Arrays vor.

Beispiel:

String[] „unkomprimiert“

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]
Z	Z	Z	Z	7	7	7	7	7	7	7	7	7	7	M	P	P	P	P	P	H	H

String[] „komprimiert“

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]
Z	Z	Z	Z	%	10	7	M	%	5	P	H	H

ZZZZ7777777777MPPPPPHH (unkomprimiert)

ZZZZ%107M%5PHH (komprimiert)

Folgende Funktionen stehen zu Verfügung:

laenge(String[] ) : Integer	Gibt die Länge des übergebenen Zeichenkettenarrays als ganze Zahl zurück
add(String[], String) : String[]	Verlängert das übergebene Zeichenkettenarray um den übergebenen String und gibt es zurück
toString(Integer) : String	Gibt die übergebene ganze Zahl als String zurück

b) Bei der Dekomprimierung von „%53V“ tritt ein Problem auf.

ba) Beschreiben Sie kurz das Problem.

2 Punkte

---



---



---



---

bb) Beschreiben Sie kurz eine mögliche Lösung.

3 Punkte

---



---



---



---

## Lösung

ba) 2 Punkte

Dekompression von %53V nicht eindeutig. Kann fünfmal 3 und einmal V oder 53-mal V bedeuten.

bb) 3 Punkte

Z. B.

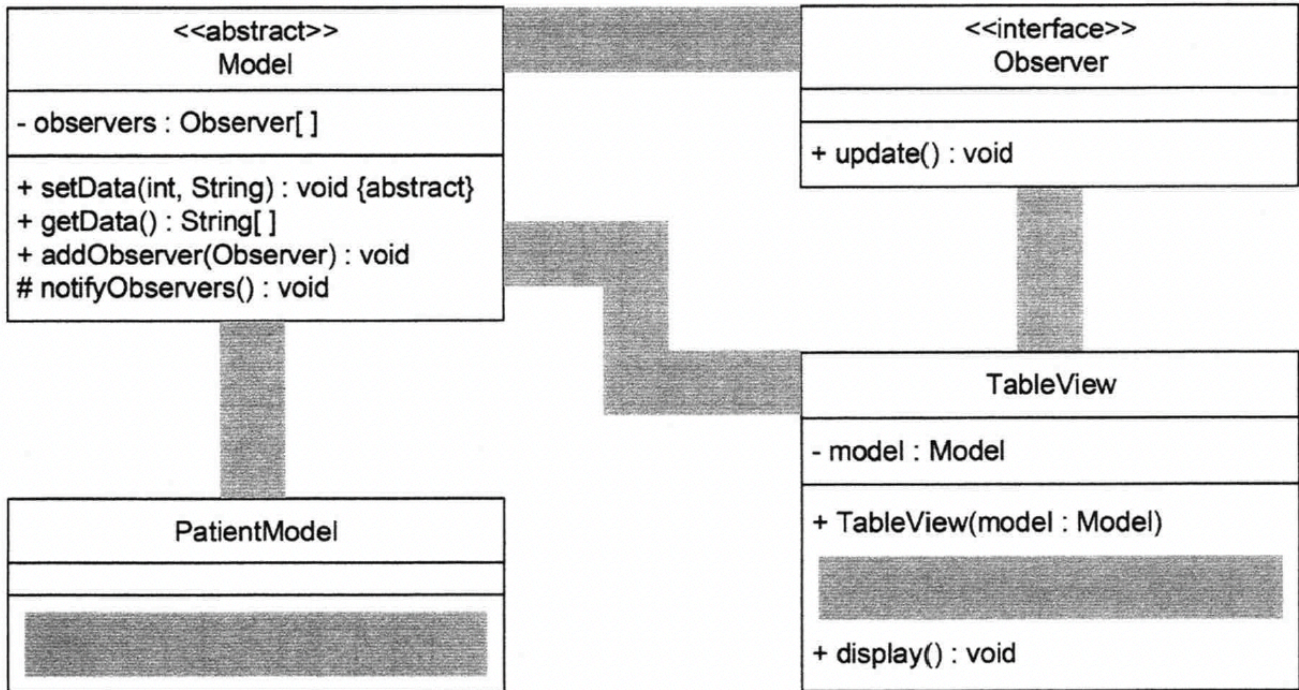
- Erkennung der Lauflängenkodierung mit Start- und Endkennzeichen
- Anzahl der Ziffern soll immer einstellig sein

## Aufgabe 24

ba) Model und View werden häufig über das Observer-Pattern realisiert. Dabei erbt die konkrete Klasse „PatientModel“ von der abstrakten Klasse „Model“. Die Klasse „TableView“ implementiert das Interface „Observer“.

Ergänzen Sie im vorliegenden UML-Klassendiagramm Methoden und Klassenbeziehungen.

6 Punkte



bb) Der Konstruktor von „TableView“ initialisiert seine Modelreferenz mit dem übergebenen Modelobjekt und registriert sich mit der Methode „addObserver“ als Observer.

Geben Sie den Konstruktor in Pseudocode an.

3 Punkte

---



---



---



---



---

bc) Die Methode „notifyObservers“ sorgt dafür, dass alle registrierten Observer die Methode „update“ ausführen.

Geben Sie die Methode in Pseudocode an.

3 Punkte

---



---



---



---

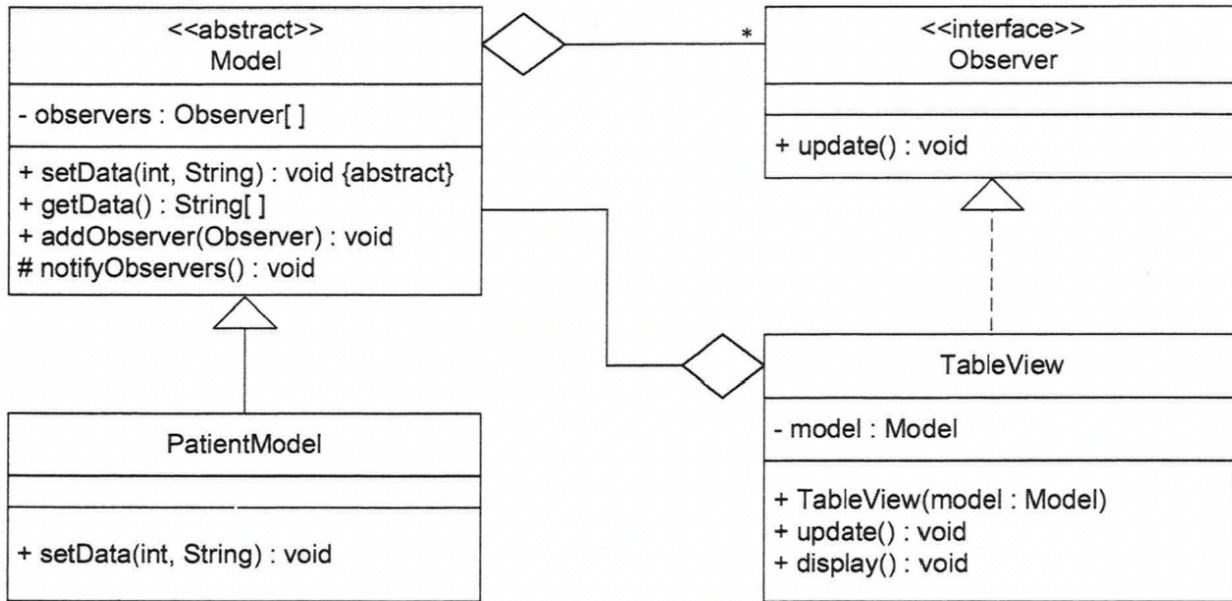


---

# Lösung

ba) 6 Punkte

1 Punkt Vererbung, 1 Punkt Realisierung, 2 x 1 Punkt je Aggregation, 2 x je 1 Punkt für überschreiben von „setData“ und „update“



bb) 3 Punkte

```

+ TableView(model : Model)
this.model := model 1 Punkte
model.addObserver(this) 2 Punkte
    
```

bc) 3 Punkte

```

# notifyObservers() : void
foreach (Observer o : observers) 2 Punkte
o.update() 1 Punkt
end foreach
    
```