



# Skript - Grundlagen der Programmierung



## Inhaltsverzeichnis

1. Einführung.....	2
1.1 Big Picture des Faches Anwendungsentwicklung und Programmierung.....	2
1.2 Entwicklung von Programmiersprachen .....	3
1.3 Programmiersprachenuniversum .....	4
1.4 Eigene Programmiersprachenkompetenz .....	5
1.5 Desktop- vs. Webapplikationen.....	6
2. Grundlagen.....	8
2.1 Teamfindungsphase.....	8
2.2 Hello World .....	8
2.2 Übersetzungsarten .....	9
2.3 Hello World .....	10
2.4 Variablen.....	10
3. Objektorientierte Programmierung .....	11
3.1 Einführung in die Objektorientierung.....	11
3.2 Einfache Klassen .....	13
3.3 Methoden .....	17
3.4 Programmverzweigungen.....	19
3.5 Schleifen .....	27
3.6 Komplexe Datentypen .....	29
3.7 Zusammenfassende Übungen .....	34
3.8 Reflexionsphase .....	36
4. Vertiefung der objektorientierten Analyse und Design .....	37
4.1 Beziehungen zwischen Klassen.....	37
4.2 Exkurs: Strukturiung von Programmcode .....	39
4.3 Vererbung .....	42
4.3.1 Grundlagen .....	42
4.3.2 Methoden in der Vererbungsstruktur .....	47
4.3.4 Diamondproblem .....	52
4.5 Interface.....	53
5. Entwurfsmuster.....	58
5.1 Strategy Pattern.....	58
5.2 Weitere Design Pattern .....	65
6. Allerlei – was noch fehlte .....	66

## 1. Einführung

### 1.1 Big Picture des Faches Anwendungsentwicklung und Programmierung

A large rectangular grid with a light gray background and a fine, uniform grid pattern, intended for drawing or taking notes.

## 1.2 Entwicklung von Programmiersprachen



Programmiersprachen haben sich im Verlauf der Zeit stetig weiterentwickelt. Allen Programmiersprachen ist jedoch gemein, dass diese letztlich in Maschinensprache übersetzt werden müssen, so dass der Computer, genauer betrachtet das Rechenwerk in Kooperation mit dem Steuerwerk, die gewünschten Befehle verstehen und ausführen können.

Leider sind folgende Generationen von Programmiersprachen mit den jeweiligen Erklärungen durcheinandergeraten. Ihre Aufgabe ist es daher, die jeweilige Generation mit der jeweilig passenden Erklärung durch eine Linie zu verbinden.

1. Generation:

Maschinensprache

Sprachen, die für ein bestimmtes Anwendungsgebiet (z. B. Datenbanksprache SQL) geschaffen wurden.

Z. B. `Select * FROM kunde`

2. Generation:

Assemblersprachen

Befehle auf dieser Ebene werden Maschinenbefehle genannt, z. B.

`1011 0011 0000 0100`, was bedeutet: (Übertrage den Wert 4 in Register BL)

3. Generation:

Höhere Programmiersprachen

Ziel der Objektorientierung ist es, ein Abbild der realen Welt vorzunehmen. Es wird davon ausgegangen, dass die Welt als Objekte, die miteinander in Beziehung stehen, abgebildet werden kann. Z.B. `Kunde klaus = new Kunde("Klaus")`, z.B. C#, Java

4. Generation:

Deskriptive= Beschreibende  
Programmiersprachen

Weiterentwicklung zu Assemblersprache, die mnemotechnische Befehle benutzt. Z.B.

`move BL, 4`      `add BL, 5`

5. Generation:

Wissens- und objektorientierte  
Programmiersprachen

Stärker an der menschlichen Sprache angelehnte Bezeichnungen für Befehle, ermöglichen eine Konzentration des Programmierers auf die eigentliche zu bewältigende Aufgabe, z. B.

`Summe = 4 + 5`, z. B. die Sprache C

### 1.3 Programmiersprachenuniversum



**Arbeitsauftrag 1:** Erstellen Sie in Ihrer Gruppe auf einer PowerPoint Folie eine Übersicht über die von Ihnen gewählte Programmiersprache. Markieren Sie Ihre Wahl, so dass die nächste Gruppe erkennen kann, dass Sie diese Sprache bereits gewählt haben.

Die Folien sollen einem Plakat entsprechen und einerseits mögliche Einsatzszenarien darstellen sowie Vor- und Nachteile nennen. Gerne können Sie auch ein kleines Codefragment oder einen Screenshot in Ihre Folien integrieren. Achten Sie hierbei darauf, die Daten grafisch anspruchsvoll, gerne auch aufgelockert, darzustellen. Bereiten Sie sich zudem darauf vor, Ihre Folien Ihren Mitschülern später auch erklären zu können.

Drucken Sie Ihre Folie anschließend aus und heften Sie es diese an der Wand/Tafel bei dieser Station an. Versuchen Sie, diese Aufgabe innerhalb von ca. 20 Minuten zu erledigen.

**Sprachen/Techniken:**

<input type="radio"/> Java	<input type="radio"/> XML	<input type="radio"/> CSS
<input type="radio"/> C#	<input type="radio"/> ColdFusion	<input type="radio"/> GO
<input type="radio"/> C++	<input type="radio"/> Ruby	<input type="radio"/> ABAP
<input type="radio"/> Java Script	<input type="radio"/> Python	<input type="radio"/> _____
<input type="radio"/> Typescript	<input type="radio"/> PHP	<input type="radio"/> _____
<input type="radio"/> SQL	<input type="radio"/> HTML	<input type="radio"/> _____

**Arbeitsauftrag 2: Diskutieren Sie in Ihrer Gruppe**, welche Sprache Ihrer Meinung die beste Programmiersprache ist. Legen Sie sich bitte passende Argumente zurecht, dies anschließend vor der Klasse zu begründen.



**Arbeitsauftrag 3:** Schauen Sie zudem nach, ob Sie Ihre gewählte Sprache im *Tiobe Index* wiederfinden. Natürlich recherchieren Sie dabei kritisch, was der *Tiobe Index* überhaupt ist und notieren Ihre Erkenntnis im nachfolgenden Feld.

<https://www.tiobe.com/tiobe-index/>



_____
-------



**Arbeitsauftrag 4: Programmiersprachenpuzzle:**

URL: <https://learningapps.org/watch?v=puzr48tyn20>



## 1.4 Eigene Programmiersprachenkompetenz

Es ist unbedeutsam mit welcher Programmierkompetenz Sie in Ihr Berufsleben starten. Die Frage ist, mit welcher Kompetenz Sie Ihre Ausbildung beenden. Bitte beantworten Sie zu diesem Zweck folgende Fragestellungen und Aufgaben.

**1. Schätzen Sie** nun Ihre aktuelle eigenen **Programmierkompetenz** ein.  
Benutzen Sie hierzu folgende Kompetenzstufen  
(**0**=keine Kenntnisse, **1**=wenig Kenntnisse, **2**=Grundlagenkenntnisse, **3**=Erweiterte Grundlagen, **4**= Profikenntnisse)

**2. Welche weiteren Ziele** im Programmierumfeld haben Sie noch?

**3. Wie** wollen Sie diese Ziele erreichen?

Programmiersprache	aktuell	Mitte der Ausbildung	Ende der Ausbildung

**4. Zeigen** Sie mit einem **Mitschüler** Ihrer Wahl Ihre Planung und bitten Sie diesen um Feedback zu Ihren Plänen.

**5. Besprechen** Sie Ihre Ziele mit Ihrem **Lehrer**. Halten Sie die Ergebnisse beider Besprechungen, die Sie für sinnvoll erachten, im nachfolgenden Feld fest.



--

## 1.5 Desktop- vs. Webapplikationen

**Arbeitsauftrag:** Nutzen Sie die bereit gestellten Puzzleteile, um das klassische, etwas vereinfachte *Zusammenspiel von HTML, CSS, PHP, SQL und JavaScript* darzustellen. Übertragen Sie anschließend das fertige Puzzlebild auf folgende Skizze.



1. Benutzer gibt Ziel-URL im Browser ein.
2. Browser fordert unter Nutzung des HTTP-Protokolls(GET-Befehl) eine Webseite bei einem Webserver an.
3. Webserver nimmt die Anfrage über seinen Port 80 wahr und führt mittels des PHP Interpreters das angeforderte PHP-Skript aus.
4. PHP-Skript fordert vom Datenbankserver mittels SQL-Befehle Daten aus der Datenbank an.
5. Datenbankserver gibt die angeforderten Daten an Webserver zurück.
6. PHP-Skript verarbeitet die empfangenen Daten vom Webserver.
7. Webserver sendet die vorbereitete Webseite mittels des HTTP-Protokolls(Response).
8. Browser visualisiert die Webseite, indem dieser HTML, CSS und JavaScript ausführt.
9. Benutzer bekommt das Resultat angezeigt und kann wiederum reagieren.

**Arbeitsauftrag:** Schauen Sie sich nachfolgendes Lernvideo an und beantworten Sie dabei stichpunktartig folgende Fragen auf der Rückseite dieses Blattes.

<https://academind.com/learn/web-dev/frontend-vs-backend-vs-fullstack-development/>



1. Mit welchen Techniken beschäftigt sich ein Frontendentwickler, mit welchen beschäftigt sich ein Backendentwickler?
  
2. Welcher Bereich pflegt den Kontakt zur Datenbank?
  
3. Welcher Bereich ist für die Validierung der Nutzereingaben zuständig?

**Arbeitsauftrag:** Recherchieren Sie im Internet was ist unter einer „Mobile App“ und was unter einer „Desktop App“ zu verstehen ist. Veranschaulichen Sie Ihrer diesbezüglichen Überlegungen im nachfolgenden Feld.



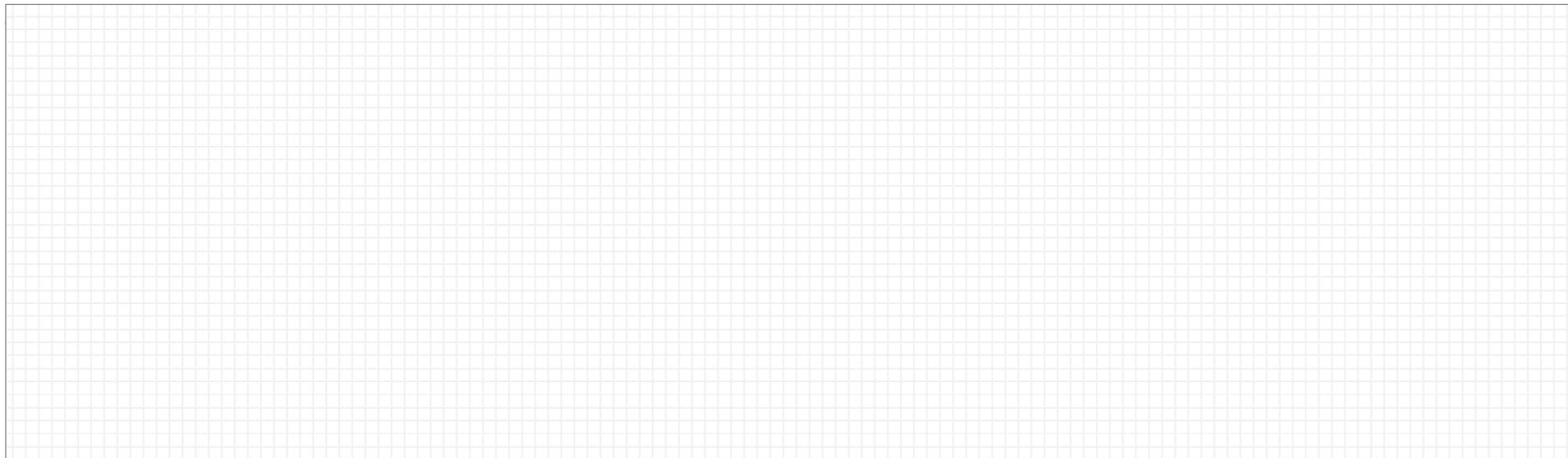
## 2. Grundlagen

### 2.1 Teamfindungsphase

**Konzeptbeschreibung:** Nach mehreren Jahren des Programmierunterrichts ist bei mir die Erkenntnis gereift, auch die Einführung in das Programmieren auf einer möglichst konzeptionellen Ebene zu unterrichten. Entsprechend dieser Überlegung wird in diesem Skript nicht eine, sondern mehrere Programmiersprachen exemplarisch benutzt, um grundlegende Konzepte zu vermitteln und vergleichend betrachten zu können.

Zu jeder dieser Programmiersprachen wird es ein entsprechendes GitHub Repository geben, dass die Realisierung der Übungsaufgaben in der jeweiligen Sprache aufzeigt. Sie sind herzlich dazu eingeladen, dieses Repository zu nutzen und zu befüllen. In diesem Skript werden Sie insofern eher selten konkreten Code sehen, da Sie diesen im GitHub Repository einsehen und herunterladen können.

 Im Anhang dieses Skriptes liegt eine Nutzungsbeschreibung von GitHub. Bitte arbeiten Sie sich selbstständig in die Benutzung ein. Folgende **Kernsprachenteams** haben wir



 GitHub-Repos:

[https://github.com/KlaraOppenheimerSchule/Grundlagen-Programmierung\\_CSharp](https://github.com/KlaraOppenheimerSchule/Grundlagen-Programmierung_CSharp)

[https://github.com/KlaraOppenheimerSchule/Grundlagen-Programmierung\\_Python](https://github.com/KlaraOppenheimerSchule/Grundlagen-Programmierung_Python)

[https://github.com/KlaraOppenheimerSchule/Grundlagen-Programmierung\\_PHP](https://github.com/KlaraOppenheimerSchule/Grundlagen-Programmierung_PHP)

[https://github.com/KlaraOppenheimerSchule/Grundlagen\\_Programmieren\\_Java](https://github.com/KlaraOppenheimerSchule/Grundlagen_Programmieren_Java)

## 2.2 Übersetzungsarten

**Arbeitsauftrag:** Schauen Sie sich nachfolgendes Lernvideo an und beantworten Sie dabei folgende Fragen:

(<https://www.youtube.com/watch?v=F1GLYZ7fhvw&feature=youtu.be>)

Um welche Fragestellung handelt es sich hierbei überhaupt? Um was geht es?



\_\_\_\_\_

Beschriften Sie die beiden nachfolgenden Elemente und beschreiben Sie deren Eigenschaften.



 \_\_\_\_\_  \_\_\_\_\_

Klären Sie nun, welchen Übersetzungsprozess die Programmiersprache Ihrer Kerngruppe nutzt und beschreiben Sie diesen im nachfolgenden Feld.



\_\_\_\_\_

## 2.3 Hello World

Richten Sie nun auf dem lokalen Schulrechner bzw. Ihrem mitgebrachten Laptop eine Entwicklungsumgebung ein und schreiben Sie ein erstes „Hello World“. Stellen Sie das Programm inklusive einer *readmeDE.md*-Datei (Beschreibt grob die Einrichtung der IDE) im GitHub Repository Ihrer Kernsprache ein. Benutzen Sie hierbei das von der Schule bereitgestellte Repository.



## 2.4 Variablen

**Arbeitsauftrag:** Recherechieren Sie für Ihre Kernsprache, welche Datentypen es gibt und ob es sich bei Ihrer Sprache um eine stark- oder schwach typisierte Sprache handelt.

Notieren Sie die Antwort im nachfolgenden Feld:

### 3. Objektorientierte Programmierung

#### 3.1 Einführung in die Objektorientierung



Alan Kay (Erfinder der ersten OOP-Sprache Smalltalk) zur Objektorientierung sagte:

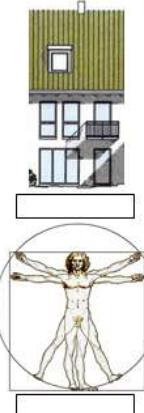
- „1. Alles ist ein Objekt
- 2. Objekte kommunizieren durch das Senden und Empfangen von Nachrichten (welche aus Objekten bestehen),
- 3. Objekte haben ihren eigenen Speicher (strukturiert als Objekte),
- 4. Jedes Objekt ist die Instanz einer Klasse (welche ein Objekt sein muss),
- 5. Die Klasse beinhaltet das Verhalten aller ihrer Instanzen (in der Form von Objekten in einer Programmliste),
- 6. Um eine Programmliste auszuführen, wird die Ausführungskontrolle dem ersten Objekt gegeben und das Verbleibende als dessen Nachricht behandelt“

**Arbeitsauftrag:** Besprechen Sie in Ihrer Kerngruppe, was Alan Kay damit sagen wollte und welche Folgerung dies für das Programmieren hat. Halten Sie Ihre Erkenntnisse stichpunktartig im folgenden Feld fest.



--

**Aufgabe:** Verbinden Sie mit einer Linie die konkreten Objekte mit ihrer Klassenvorlage. Notieren Sie jeweils unter dem Bild, ob es sich um ein Objekt oder um eine Klasse handelt.



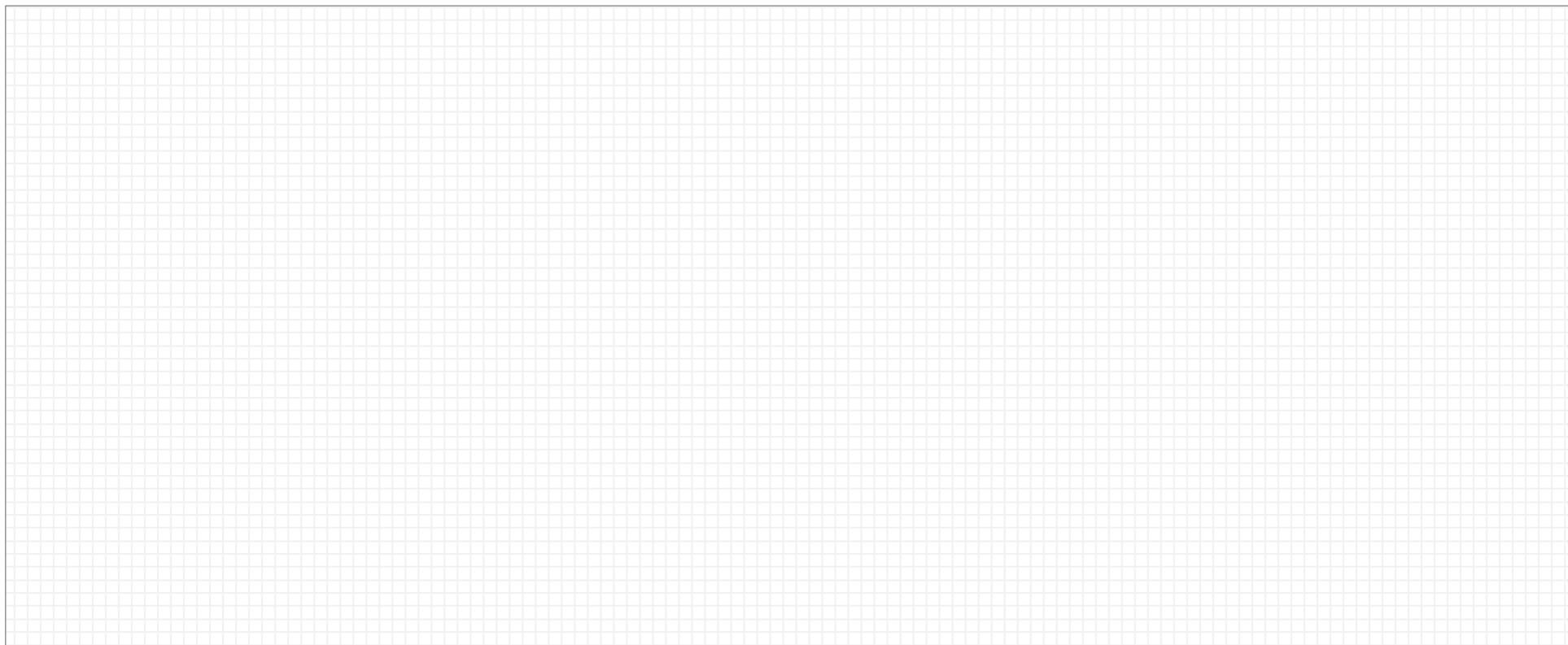
--

**Aufgabe:** Überlegen Sie sich, welche Klassen es in Ihrer Lieblingsberufsschule gibt.

A large rectangular grid of light gray squares, intended for students to write their answers to the assignment question.

### 3.2 Einfache Klassen

Anwendungskontext der einführenden Beispiele: In dem nachfolgenden Kapitel werden wir Schritt für Schritt eine digitale Verwaltung für eine Buchhandlung aufbauen. Beginnen werden wir mit dem zentralen Element eines Buchhandels: Dem Buch.



Hinweis: Neben dem Objektorientierten Paradigma gibt es noch weitere Paradigmen, d.h. Denkmuster. Hinter folgenden Link erhalten Sie einen spannenden, kurzweiligen Eindruck über das Imperative, Funktionale und deklarative Paradigma. <https://t1p.de/na1d>



<pre> class Book {     private string isbn;     private string title;     private string author;     private double price;      public Book(string isbn, string title, string author, double price)     {         this.isbn =isbn;         this.title =title;         this.author =author;         this.price = price;     }      public string getIsbn()     {         return this.isbn;     }      public void setIsbn(string isbn)     {         this.isbn=isbn;     }      //... }  class Program {     static void Main(string[] args)     {         Book myFirstBook = new Book("123456789", "1984", "George Orwell", 20.4);         Console.WriteLine(myFirstBook.getIsbn());         Console.ReadLine();     } } </pre>	<pre> &lt;?php class Book {     private \$isbn;     private \$title;     private \$author;     private \$price;      public function __construct(string \$isbn, string \$title, string \$author, float \$price)     {         \$this-&gt;isbn=\$isbn;         \$this-&gt;title=\$title;         \$this-&gt;author=\$author;         \$this-&gt;price=\$price;     }      public function getIsbn(): string     {         return \$this-&gt;isbn;     }      public function setIsbn(string \$isbn): void     {         \$this-&gt;isbn=\$isbn;     } }  \$myFirstBook=new Book("123456789", "1984", "George Orwell", 20.6); echo \$myFirstBook-&gt;getIsbn(); </pre>
---	--

<pre> class Book:     def __init__(self, isbn, title, author, price):         self.__isbn = isbn         self.__title = title         self.__author = author         self.__price = price      def getIsbn(self):         return self.__isbn      def setIsbn(self, isbn):         self.__isbn = isbn  buch = Book("123456789", "1984", "George Orwell", 20.6) print(buch.getIsbn()) </pre>	
---	--

**Arbeitsauftrag:** Vergleichen Sie nun die verschiedenen Programmiersprachen zueinander. Notieren Sie **Unterschiede** und **Gemeinsamkeiten**.

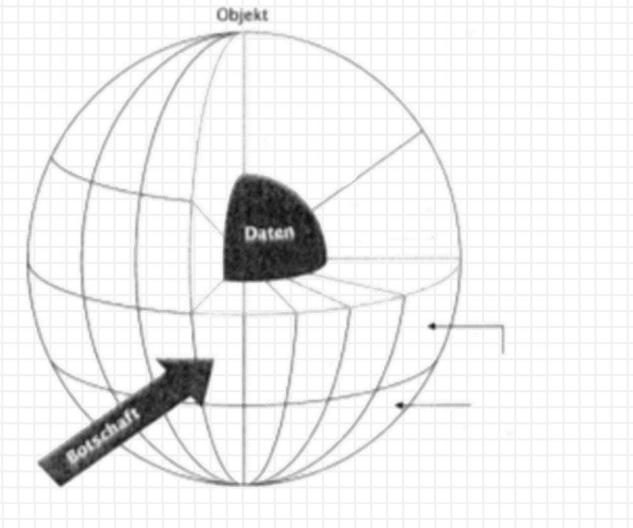


### Sprachenvergleich

- Jede Klasse sollte in einer eigenen Datei mit dem gleichen Namen wie die Klasse geschrieben werden.
- Klassennamen sollten CamelCased sein, d.h. jedes TeilWort sollte mit einem Großbuchstaben beginnen, der Rest dann in Kleinbuchstaben
- Eine Datei sollte nur den Code für eine Klasse enthalten.
- In einer Klasse sollten zunächst die Attribute, dann der Konstruktor und dann die restlichen Methoden gesetzt sein.

**Datenkapselung:**

Wenn man mit Objekten arbeitet, ist eines der wichtigsten Konzepte das der Kapselung. Kapselung versucht, die interne Implementierung nach außen hin zu verstecken. Alle Attribute sollten deshalb private sein; der Zugriff sollte nur über public-Methoden möglich sein. Der Grund für dieses Prinzip ist, dass es dem Entwickler leichter gemacht werden soll, Änderungen an der internen Struktur durchzuführen zu können, ohne den Code beeinflussen zu müssen, der auf diese Attribute zugreift.



**Transferaufgabe zu Klassen und Objekten:**

Einem Nutzer soll es möglich sein, basierend auf dessen Eingaben, einen Mitarbeiter auf dem Server anzulegen. Die vom Nutzer übermittelten Daten (Nachname, Vorname) werden auf dem Server genutzt, um dort basierend auf einer Klasse Mitarbeiter jeweils das konkrete Mitarbeiterobjekt zu erzeugen. Der übermittelte Vorname und der Nachname sollen im Rahmen des Konstruktoraufrufes übergeben und im Mitarbeiterobjekt jeweils als Attribut gespeichert werden.

Anschließend sollen der Vorname und der Nachname des erzeugten Mitarbeiterobjektes, basierend auf einem Aufruf der entsprechenden getter-Methoden der Mitarbeiter am

Bitte tragen Sie die Daten des zu erzeugenden Mitarbeiters ein

Mein Vorname ist: Christoph

Mein Nachname ist: Zobel

Bildschirm ausgegeben werden. Das Programm könnte wie folgt aussehen:

Sollten Sie die Aufgabe mit einem anderen Sprache umsetzen, die nicht webbasiert ist, können Sie die Ein- und Ausgabe in der Konsole vornehmen.

### 3.3 Methoden

Methoden werden innerhalb einer Klasse definiert. Ihr Vorteil liegt darin, dass sie die Eigenschaften der jeweiligen Objekte nutzen können.

**Arbeitsauftrag:** Doch aus welchen Elementen besteht eigentlich eine Methode: Erstellen Sie im folgenden Feld ein kurzes Mindmap hierzu:

A large rectangular grid area for drawing a mindmap. It is intended for students to draw a hierarchical diagram showing the elements of a method.

**Übungen zu Methoden:** *Hinweis: Erstellen Sie alle nachfolgenden Methoden in einer Klasse Mathe, von der Sie ein Objekt erzeugen. Die Aufgaben 1-4 können Sie in einer Klasse mit den Klassennamen Mathe zusammenfassen.*

1. Matheaufgaben:
  - a. Erstellen Sie eine Methode, die zwei Zahlen übergeben bekommt und die Summe dieser Zahlen zurückgibt. Die Ausgabe dieser Summe soll nach erfolgtem Aufruf der Methode vom Aufrufer der Methode ausgegeben werden. (BA)
  - b. Eine Methode erhält eine Zahl und gibt die Quadratzahl aus. (BA)
  - c. Eine Methode soll einen int-Wert erhalten und überprüfen, ob dieser positiv ist. Sofern der Wert positiv ist, soll die Methode ein true, andernfalls ein false an den Aufrufer zurückgeben. (BA)
  - d. Programmieren Sie eine Methode, die die Fakultät von einer übergebenen natürlichen Zahl errechnet und an den Aufrufer zurückgibt. Beispiel  $3!=1*2*3=6$ .  
*Erweiterung für Schnelle:* Realisieren Sie obige Funktion rekursiv. (DS)
2. Programmieren Sie die Türme von Hanoi. Sprechen Sie mich bezgl. des Visualisierungsmaterial bitte an. (SKMNBT)

## Fragen

**F:** Was sind überladene Methoden?

**A:** Überladen (overloading) bedeutet, dass man für zwei Funktionen identische Funktionsnamen verwendet, die aber verschiedene Argumente besitzen.

**F:** Gibt es in der Programmiersprache Ihrer Kerngruppe überladenen Methoden?

**A:** \_\_\_\_\_

**F:** Worin liegt der Unterschied zwischen Methoden und Funktionen?

**A:** Eine Methode kann vereinfacht als eine Funktion bezeichnet werden, die in einer Klasse implementiert wurde.

**F:** Gibt es in der Programmiersprache Ihrer Kerngruppe Funktionen?

**A:** \_\_\_\_\_

**Arbeitsauftrag:** Vergleichen Sie nun die Methoden der verschiedenen Programmiersprachen zueinander. Notieren Sie **Unterschiede** und **Gemeinsamkeiten**.

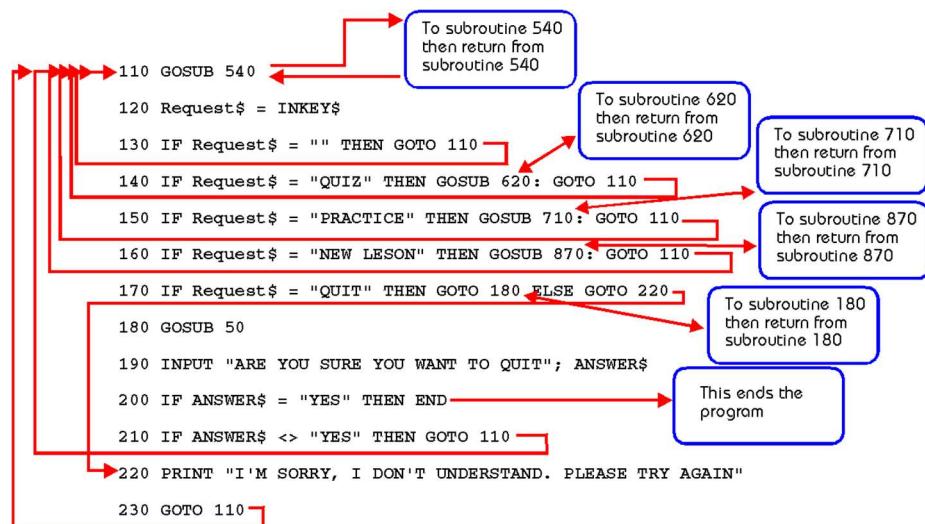


Sprachenvergleich

### 3.4 Programmverzweigungen

#### If/else-Kontrollstruktur

Kontrollstrukturen sind ein zentraler Bestandteil fast jeder Programmiersprache. Erst diese ermöglichen einen flexiblen, unterschiedlichen Fluss des Programmablaufes in Abhängigkeit der Auswertung von Variablen. Im Jahr 1972 kam Isaac Nassi und Ben Schneiderman auf die Idee, Sachverhalte der strukturierten Programmierung in Form von Diagrammen, genannt Struktogramm abzubilden. Ziel war es, weniger sogenannten Spagetticode zu produzieren. Spagetticode können Sie sich wie folgt vorstellen.



Quelle: [http://cwarden.org/warden/personalPage/caell/7\\_3.html](http://cwarden.org/warden/personalPage/caell/7_3.html)

Daher und auch weil die IHK das Thema Struktogramme immer noch spannend bei der Prüfung findet, machen wir einen kleinen Exkurs zum Thema Struktogramme.

#### Struktogramm

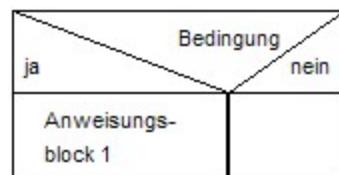
Die meisten der nachfolgenden Strukturblöcke können ineinander geschachtelt werden. Das aus den unterschiedlichen Strukturblöcken zusammengesetzte Struktogramm ist im Ganzen rechteckig, also genauso breit wie sein breitestes Strukturblock.

### Linearer Ablauf (Sequenz)

Anweisung 1	Jede Anweisung wird in einen rechteckigen Strukturblock geschrieben.
Anweisung 2	Die Strukturböcke werden nacheinander von oben nach unten durchlaufen.
Anweisung n	Leere Strukturböcke sind nur in Verzweigungen zulässig.

### Verzweigung (Alternativen)

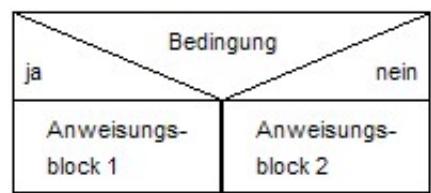
#### Einfache Auswahl



Alternativ: bedingte Verarbeitung, Selektion, einfache Selektion (if)

Nur wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Trifft die Bedingung nicht zu (falsch), wird der Durchlauf ohne eine weitere Anweisung fortgeführt (Austritt unten).

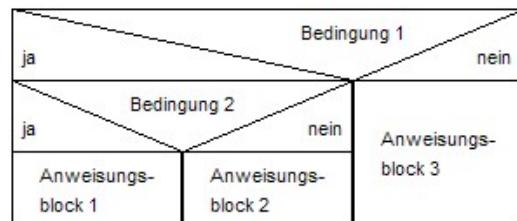
#### Zweifache Auswahl



Alternativ: alternative Verarbeitung, alternative Verzweigung (if then else)

Wenn die Bedingung zutreffend (wahr) ist, wird der Anweisungsblock 1 durchlaufen. Trifft die Bedingung nicht zu (falsch), wird der Anweisungsblock 2 durchlaufen. Ein Anweisungsblock kann aus einer oder mehreren Anweisungen bestehen. Austritt unten nach Abarbeitung des jeweiligen Anweisungsblocks.

#### Verschachtelte Auswahl

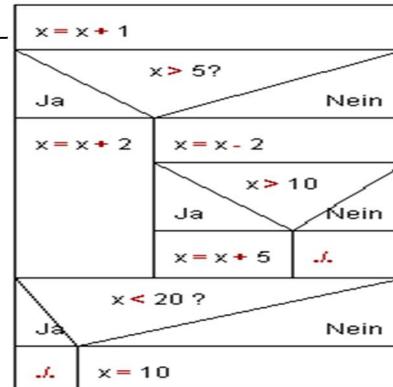


Es folgt eine weitere Bedingung. Die Verschachtelung ist ebenso im Nein-Fall (noch) möglich.

Übungen zu Struktogrammen:

1. Welchen Wert hat die Variable x nach Durchlauf des Programms, wenn

- a) X zu Beginn den Wert 1 hat
- b) X zu Beginn den Wert 17 hat

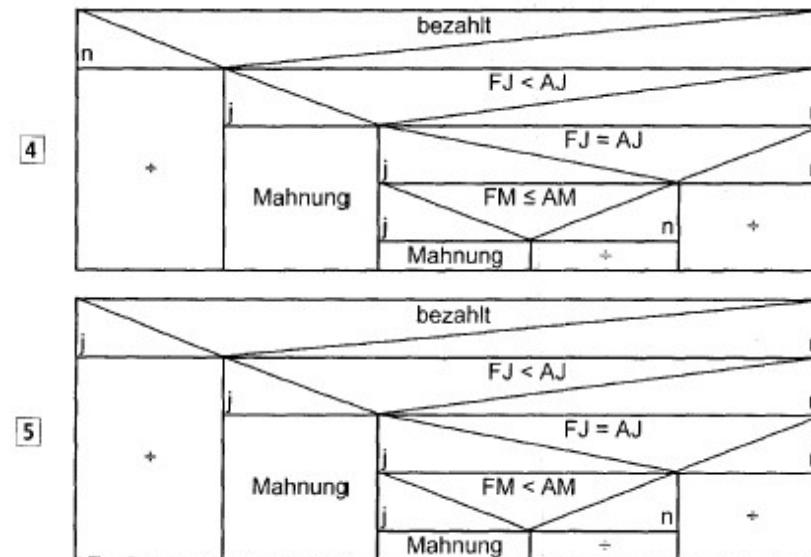
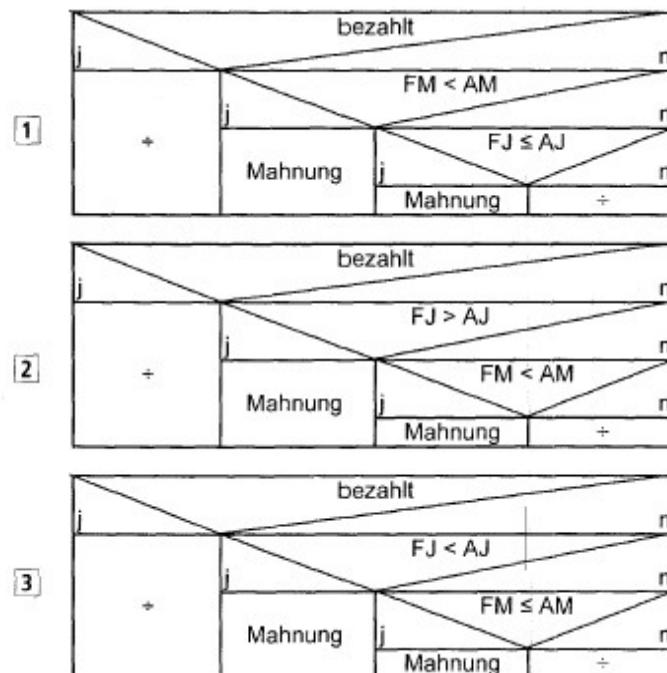


2. Für ein Mahnprogramm soll eine Verarbeitungslogik entworfen werden. Die Mahnung wird generell erst im auf den Fälligkeitsmonat folgenden Monat aufgelöst, d.h. der genaue Fälligkeitstag wird bei der Verarbeitung nicht berücksichtigt. Kontrollieren Sie, welches Struktogramm eine richtige Lösung des Problems enthält. Notieren Sie die jeweils falsche Stelle mit einem roten Stift.

Legende: AJ: Aktuelles Jahr,

FJ: Fälligkeitsjahr,

AM: Aktueller Monat, FM: Fälligkeitsmonat.



3. Im Rahmen der nachfolgenden Aufgabe sollen Sie einen so genannten Schreibtischtest durchführen. Es stellt sich dabei zwangsläufig die Frage, was ein Schreibtischtest überhaupt ist. Fragen wir hierzu einen einschlägig bekannten Experten, erhalten wir folgende Antwort:

*„Der Schreibtischtest ist ein Verfahren, das im Bereich der Softwareentwicklung verwendet wird, um Algorithmen oder Routinen auf Richtigkeit zu prüfen. Der Schreibtischtest wird nicht mit Hilfe eines Rechners durchgeführt, sondern vielmehr im Kopf des Entwicklers“, das zumindest habe ich auf Wikipedia gelesen.*



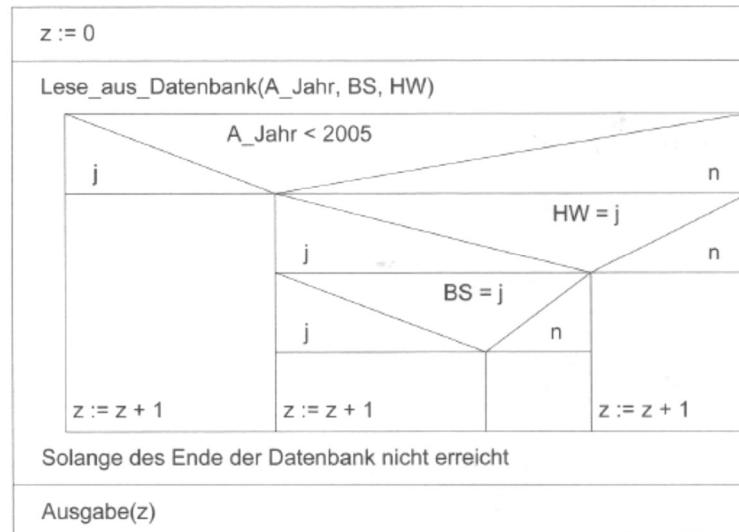
Im neuen Netzwerk der Schwarzenbach KG sollen alle Arbeitsplatz-PCs einheitlich mit einem neuen Betriebssystem betrieben werden und es sollen auch noch alte Arbeitsplatz-PCs weiterverwendet werden.

Zur Ermittlung der benötigten Anzahl Lizenzen soll ein entsprechendes Programm erstellt werden. Ein Mitarbeiter der IT-Abteilung hat bereits folgendes Struktogramm erstellt.



Hinweis

- Das Programm greift auf die Hardware-Verwaltungs-Datenbank zu.
- HW: Hardware-Ausstattung des PCs erfüllt Anforderung (j oder n)
- A\_Jahr: Jahr, in dem PC angeschafft wurde
- BS: Installiertes Betriebssystem neu (j oder n)



Testen Sie die im Struktogramm dargestellte Logik mit einem Schreibtischtest anhand folgender Daten:

A_Jahr	BS	HW
2004	n	j
2004	n	n
2006	n	n
2007	n	j
2008	j	j
2009	j	j
2009	j	j

Wie viele Lizenzen müssen laut Ihrem Schreibtischtest beschafft werden?

C#	<pre>... public double calculateShippingCoasts() {     if(this.price&gt;20)     {         return 0;     }     else     {         return 3.90;     } }</pre>	PHP	<pre>... public function calculateShippingCoasts() : float {     if(\$this-&gt;price&gt;20)     {         return 0;     }     else     {         return 3.90;     } }</pre>
----	---	-----	---

Python	<pre>... def calculateShippingCoasts(self):     if(self.__price &gt; 20):         return 0     else:         return 3.90</pre>		
--------	--	--	--

## Switch-Kontrollstruktur

Eine Kontrollstruktur ähnlich zu if..else ist switch ... case. Diese Struktur wertet nur **einen Ausdruck** aus und führt einen Block in Abhängigkeit seines Wertes aus.

C#	<pre>public String getBookKategorie() {     string categoryName;      switch(this.category)     {         case 1:             categoryName= "Action";             break;         case 2:             categoryName = "Fantasy";             break;         case 3:             categoryName = "Horror";             break;         case 4:             categoryName = "Classics";             break;         default:             categoryName = "Unknown";             break;     }     return categoryName; }</pre>	PHP	<pre>public function getBookKategorie() : string {     \$categoryName;      switch(\$this-&gt;category)     {         case 1:             \$categoryName= "Action";             break;         case 2:             \$categoryName = "Fantasy";             break;         case 3:             \$categoryName = "Horror";             break;         case 4:             \$categoryName = "Classics";             break;         default:             \$categoryName = "Unknown";             break;     }     return \$categoryName; }</pre>
----	--	-----	--

Python			
--------	--	--	--

**Arbeitsauftrag:** Vergleichen Sie nun die verschiedenen Programmiersprachen zueinander. Notieren Sie **Unterschiede** und **Gemeinsamkeiten**.



**Sprachenvergleich**

---

## Fragen

**F:** Muss jede Switch-Kontrollstruktur auch einen Default-Fall angeben?

**F:** Benötige ich nach jeder Anweisung den Befehl break?

**A:** Es ist auch wichtig zu wissen, dass die break-Anweisungen notwendig sind, wenn man den switch-case-Block verwenden will. Ohne break wird PHP die weiteren Cases ausführen, auch wenn dies nicht gewünscht ist.

**A:** Nein. Die Angabe eines Defaultfalles ist optional und je nach Situation sinnvoll oder auch nicht.

**F:** Was ist das Zeichen in der Programmiersprache Ihrer Gruppe für folgende logische Operatoren:

**A:** oder = \_\_\_\_\_

**A:** ungleich = \_\_\_\_\_

## Übungspool zu Kontrollstrukturen:

- Erstellen Sie nun ein Programm, das einen Nutzer dabei hilft, dessen Kosten für einen Tankvorgang zu berechnen. Der Nutzer soll hierzu die Anzahl der getankten Liter eingeben sowie die getankte Sorte. Anschließend dargestellt werden. Sollte die getankte Menge größer oder gleich 100 Liter sein, wird nochmals 5% Rabatt vom Preis abgezogen.
- Erstellen Sie ein Programm, das die erreichten IHK-Punkte (siehe Abbildung rechts), in Noten umrechnet und ausgibt. Sollten ein unpassender Wert eingegeben werden, wird der Nutzer darauf aufmerksam gemacht.
- Für ein Unternehmen sollen wir ein Programm entwickeln, welches für erfasste LKW automatisch die entsprechende LKW-Mautgebühr berechnet.

Folgende Mautgebühren wurden uns dafür vorgegeben:

Schadstoffklasse	A	B	C	D	E	F
bis 3 Achsen	12,50 ct/km	14,60 ct/km	15,70 ct/km	18,80 ct/km	19,80 ct/km	20,80 ct/km
ab 4 Achsen	13,10 ct/km	15,20 ct/km	16,30 ct/km	19,40 ct/km	20,40 ct/km	21,40 ct/km

Aufgabenstellung: Schreiben Sie eine Methode, die nach der oben beschriebenen Regelung die LKW-Maut für eine Fahrt berechnet. Die Methode erhält sowohl die Schadstoffklasse, die Anzahl der Achsen als auch die gefahrenen Kilometer zur Berechnung übergeben. Die Mautgebühr ergibt sich unter anderem aus dem Kilometerpreis mal der Anzahl der gefahrenen Kilometer.<sup>1</sup>

- Erstellen Sie eine Methode, die prüft, ob ein Schaltjahr vorliegt. Schaltjahre liegen dann vor, wenn die Jahreszahl durch 4 teilbar ist, jedoch nicht zusätzlich durch 100 teilbar ist. Ist die Jahreszahl jedoch durch 400 teilbar, liegt wiederum ein Schaltjahr vor. Nachdem ein Nutzer eine Jahreszahl eingegeben hat, soll im Backend in einer Methode aufgerufen werden, die prüft, ob ein Schaltjahr vorliegt. Die Funktion gibt *true* oder *false* zurück. Entsprechend der Rückgabe wird der Nutzer informiert. *Erweiterung für Schnelle*: Überlegen Sie sich, wie man die Richtigkeit der Implementierung idealer weise testen könnte. Würde hier TDD weiterhelfen können?<sup>2</sup>

<sup>1</sup> Vgl. Programmierentrainieren, Lolacono/Wiefling/Schneider, 2018.

<sup>2</sup> Vgl. Programmierentrainieren, Lolacono/Wiefling/Schneider, 2018.

IHK-Notenschlüssel (dezimal)

Punkte	Schulnote
100	1,0
99	1,1
98	1,1
97	1,2
96	1,2
95	1,3
94	1,3
93	1,4
92	1,4
91	1,5
90	1,6
89	1,7
88	1,8
87	1,9
86	2,0
85	2,0
84	2,1
83	2,2
82	2,3
81	2,4
80	2,5
79	2,6
78	2,7
77	2,7
76	2,8
75	2,9
74	2,9
73	3,0
72	3,1
71	3,1
70	3,2
69	3,3
68	3,3
67	3,4
66	3,5
65	3,6
64	3,6
63	3,7
62	3,7
61	3,8

Punkte	Schulnote
60	3,9
59	3,9
58	4,0
57	4,0
56	4,1
55	4,1
54	4,2
53	4,3
52	4,3
51	4,4
50	4,4
49	4,5
48	4,6
47	4,6
46	4,7
45	4,7
44	4,8
43	4,8
42	4,9
41	4,9
40	5,0
39	5,0
38	5,0
37	5,1
36	5,1
35	5,2
34	5,2
33	5,3
32	5,3
31	5,4
30	5,4
29	5,5
23 - 28	5,6
17 - 22	5,7
12 - 16	5,8
6 - 11	5,9
0 - 5	6,0

100 - 92 Punkte	sehr gut
unter 92 - 81 Punkte	gut
unter 81 - 67 Punkte	befriedigend
unter 67 - 50 Punkte	ausreichend
unter 50 - 30 Punkte	mangelhaft
unter 30 - 0 Punkte	ungenügend

### 3.5 Schleifen

Bereiten Sie eine kurze Schulung vor, in der Ihre Mitschüler die Ihnen zugeteilte Wiederholungsstruktur in Ihrer Programmiersprache kennenlernen.

Überlegen Sie sich hierbei zuerst, wie Sie diese Thematik Ihren Mitschülern näherbringen können. Nach einer kurzen Einführung, die sowohl die Syntax als auch das Struktogramm dieser Struktur beinhalten muss, sollten Ihre Mitschüler auch eine kleine von Ihnen erstellte Aufgabe lösen können. Bitte bereiten Sie zu diesem Zwecke Ihren Mitschülern auch ein Handout vor.

Bearbeitungszeit: 45 Minuten, Präsentations- und Übungszeit: 15 Minuten

**Arbeitsauftrag:** Vergleichen Sie nun die verschiedenen Programmiersprachen zueinander. Notieren Sie **Unterschiede** und **Gemeinsamkeiten**.

Sprachenvergleich

#### Übungen zu Schleifen:

1. Erstellen Sie unter Nutzung einer Wiederholungsstruktur eine Methode in der Klasse Mathe, die die Zahlen 1 bis 1000 ausgibt.
2. Erstellen Sie unter Nutzung einer Wiederholungsstruktur eine Methode in der Klasse Mathe, die die Zahlen 1000 bis 0 in 100er Schritten ausgibt.
3. Schreiben Sie eine Methode in der Klasse Mathe, die die von einem Nutzer eingegebene Zahl wiederholt halbiert bis diese kleiner als 0,001 ist.

4. Erzeugen Sie die Ausgabe des kleinen Einmaleins mit Hilfe einer geschachtelten For-Schleife in einer Tabelle, die wie folgt aussieht:

*Tipp:* Schleifen lassen sich auch in sich verschachteln...

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

5. Erstellen Sie ein Methode in einer Klasse, die den Nutzer fragt, wie viele PIN-Codes er erzeugt haben möchte. Erzeugen Sie daraufhin die gewünschte Anzahl von Codes und geben Sie diese aus. Gehen Sie davon aus, dass ein Code grundsätzlich vierstellig ist und nur aus Zahlen bestehen sollte.

*Mögliche Erweiterung:* Stellen Sie sicher, dass nur einmalige und keine mehrfach vorkommenden Codes zurückgeben werden. Weiterhin können Sie die Anzahl der Codestellen dynamisch vom Nutzer vorgeben lassen.

*Tipp:* Nutzen Sie zum Erzeugen der Zufallszahl folgende Funktion: <http://php.net/manual/de/function.random-int.php>

6. Erstellen Sie ein Programm (inklusive Struktogramm vorab), das den Verlauf des Wertverlustes von Investitionsgütern darstellt. Der Nutzer soll dabei den Anschaffungswert und den die Abschreibungsdauer eingeben. Basierend auf diesen Werten werden dann der Anfangswert, der Abschreibungsbetrag sowie der Restwert am Jahresende ausgegeben.  
Die Ausgabe könnte wie folgt aussehen

Nutzungsjahr	Anfangswert	Abschreibung	Restwert
1	x.fff,xx	x.fff,xx	x.fff,xx
n	x.fff,xx	x.fff,xx	x.fff,xx

Zur Info:

- Abschreibungsbetrag = Anschaffungswert/Abschreibungsdauer (für alle Jahre gleich hoch)
- Der Restwert am Jahresanfang =Anschaffungswert am Jahresanfang-Abschreibungsbetrag des Jahres

### 3.6 Komplexe Datentypen

**Arbeitsauftrag:** Welche komplexen Datentypen gibt es überhaupt und wie lassen sich diese kategorisieren? Erstellen Sie hierzu ein kurzes Mindmap hierzu.

A large rectangular grid area for drawing a mindmap. It has a thin black border and a light gray grid pattern.

Bereiten Sie eine kurze Schulung vor, in der Ihre Mitschüler den Ihnen zugeteilte komplexen Datentyp in Ihrer Programmiersprache kennenlernen.

Überlegen Sie sich hierbei zuerst, wie Sie diese Thematik Ihren Mitschülern näherbringen können. Nach einer kurzen Einführung, die sowohl die Syntax als auch das Struktogramm dieser Struktur beinhalten muss, sollten Ihre Mitschüler auch eine kleine von Ihnen erstellte Aufgabe lösen können. Bitte bereiten Sie zu diesem Zwecke Ihren Mitschülern auch ein Handout vor.

Bearbeitungszeit: 45 Minuten, Präsentations- und Übungszeit: 10 Minuten

**Arbeitsauftrag:** Vergleichen Sie nun die verschiedenen Programmiersprachen zueinander. Notieren Sie **Unterschiede** und **Gemeinsamkeiten**.



Sprachenvergleich

### Frage

**F:** Woran erkenne ich, wann die Nutzung eines eindimensionalen oder mehrdimensionalen Arrays sinnvoll ist?

**A:** Die Antwort ergibt sich auch hier aus dem realen Kontext, den Sie abbilden wollen. Liegt zum Beispiel eine Struktur im Sinne eines Weihnachtskalenders vor, lassen sich die zwei Dimensionen Tag und Inhalt bei einem Türchen erkennen. Tipp: Die Orientierung am realen Bild, das man abbilden will, erscheint eigentlich immer sinnvoll.

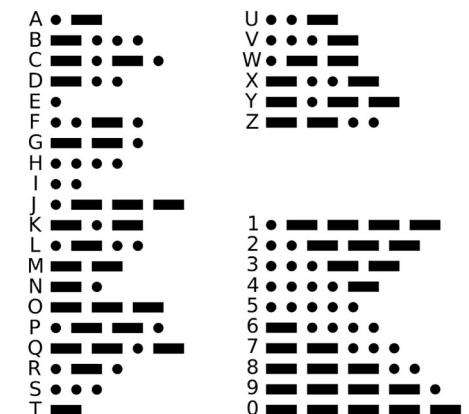


Link zu Erklärvideos zur Objektorientierten Programmierung:  
[https://klaraoppenheimer-my.sharepoint.com/:f/g/personal/zobel\\_klara-oppenheimer-schule\\_de/EpoZP22K5TNCos1WZCchPBoBlukiOVaChbcrdvBSbfuhtw?e=oIYijg](https://klaraoppenheimer-my.sharepoint.com/:f/g/personal/zobel_klara-oppenheimer-schule_de/EpoZP22K5TNCos1WZCchPBoBlukiOVaChbcrdvBSbfuhtw?e=oIYijg)



## Übungspool zu eindimensionale Datenstrukturen:<sup>3</sup>

1. Best Friends (BAF):
  - a. Erstellen Sie ein Array, das Ihre drei besten Klassenfreunde speichert und diese anschließend wieder ausgibt.
  - b. Ein Freund ist in Ungnade gefallen, löschen Sie diesen aus dem Array. Geben Sie den Inhalt des Arrays anschließend wieder aus.
  - c. Sie haben eine neue Freundin kennen gelernt, sie heißt Uschi. Speichern Sie auch diese in Ihrem Array. Geben Sie den Inhalt das Array anschließend wieder aus.
  
2. Sie wollen herausfinden, wie viele Kilometer Sie im Durchschnitt mit einer Tankfüllung fahren können. Folgende Werte wurden notiert: (1020, 923, 780, 890). (AF)
  - a. Berechnen Sie die durchschnittliche Kilometerreichweite für Ihr Fahrzeug.
  - b. Ermitteln Sie den durchschnittlichen Kraftstoffverbrauch je 100 km unter der Annahme, dass Sie einen 70 Liter Tank haben.
  
3. Erstellen Sie ein Array mit den Zahlen von 0 bis 9. Erstellen Sie anschließend **ein zweites Array** und weisen Sie diesen den Inhalt des ersten Arrays in umgekehrter Reihenfolge zu. Folgendes Beispiel soll dies simulieren.  $[0,1,2,3,4,5,6,7,8,9] \rightarrow [9,8,7,6,5,4,3,2,1,0]$ . (AF)
  
4. Erstellen Sie ein Programm, das die Ziehung der Lottozahlen simuliert. Füllen Sie zunächst ein Array mit sechs Zahlen, die zufällig über folgenden Befehl erzeugt werden: `rand(1, 49);` (DS)  
 Der Spieler kann anschließend sechs Zahlen zwischen 1 und 49 eingeben, die in einem weiteren Array gespeichert werden. Das Programm zeigt anschließend beide Zahlenreihen untereinander an. Der Abgleich der richtig erratenen Zahlen wird zunächst dem Nutzer überlassen.  
*Erweiterung für Schnelle:* Erstellen Sie eine Methode, die die Überprüfung der Zahlen selbstständig ausführt.



(Abbildung: Quelle: [https://en.wikipedia.org/wiki/Morse\\_code](https://en.wikipedia.org/wiki/Morse_code))

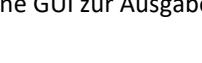
<sup>3</sup> *Pädagogischer Hinweis:* Es erscheint mir nicht sinnvoll, wenn alle Schüler alle Aufgaben am besten auch noch immer zur gleichen Zeit bearbeiten. Sinnvoller ist es, wenn jeder die Aufgabe auswählt, die seinen aktuellen Lernbedürfnissen am besten entspricht. Daher tragen die Aufgaben in Klammer am Ende des Textes Buchstabenkürzel, die Ihnen dabei helfen sollen, die Aufgabe auszuwählen, die ein passenden Schwierigkeitsniveau für Sie hat. Folgende Bedeutung steht hinter den Kürzeln (BA= Blutiger Anfänger, A= Erprobter Anfänger, DS = Durchschnitt, F = Fortgeschrittener). Sollten Sie sich keiner Gruppe zuordnen könnte es sich bei Ihnen um ein SKMNMLT handeln. SKMNMLT steht für = SchuleKannMirNichtsMehrLernenTyp. Sollten Sie diesen Typus entsprechen, bitte ich Sie auf mich zuzugehen, damit wir eine sinnvolle Lernarrangement für Sie gemeinsam finden können.

Übungspool zu mehrdimensionale Datenstrukturen:

1. Speichern Sie den Inhalt folgender Datenstruktur mit einem komplexen Datentyp Ihrer Wahl ab und geben Sie den Inhalt anschließend aus:<sup>4</sup> (BA)

Nr.	Titel	Originaltitel	Dt. Erstausstrahlung
1	Es weihnachtet schwer	Simpsons Roasting on an Open Fire	6. Dez. 1991
2	Bart wird ein Genie	Bart the Genius	20. Sept. 1991
3	Der Versager	Homer's Odyssey	11. Okt. 1991
4	Eine ganz normale Familie	There's No Disgrace Like Home	13. Sep. 1991

2. Sie erhalten Sie ehrenvolle Aufgabe, folgende Datenstruktur zu speichern und anschließend komplett auszugeben. (A)

 BRASILIEN	5	 1958 1962 1970 1994 2002
 ITALIEN	4	 1934 1938 1982 2006
 DEUTSCHLAND	4	 1954 1974 1990 2014
 URUGUAY	2	 1930 1950
 ARGENTINIEN	2	 1978 1986
 ENGLAND	1	 1966
 FRANKREICH	1	 1998
 SPANIEN	1	 2010

Erweiterung für Schnelle: Nutzen Sie eine GUI zur Ausgabe (z.B. im Webumfeld das HTML-Element table um eine ansprechende Ausgabe zu erzeugen)

---

<sup>4</sup> Quelle: <http://de.simpsons.wikia.com/wiki/Episodenguide>

3. Erstellen Sie für einen Kinosaal ein Reservierungssystem (stark vereinfacht), das es ermöglicht, freie Plätze je nach Kundenwunsch zu reservieren. Da der jetzige Kenntnisstand noch keine Nutzung einer persistenten Quelle vorsieht, werden wir uns vorerst damit begnügen, dass der Nutzer folgenden Ablauf vorfindet: (DS/F)

Nach der Eingabe der gewünschten Reihe sowie der Platznummer wird geprüft, ob der gewünschte Platz aktuell verfügbar ist. Ob ein Platz in dem erstmals kleinen Kinosaal mit drei Reihen und je fünf Plätzen verfügbar ist, kann in einem mehrdimensionalen komplexen Datentyp erkannt werden. Der mehrdimensionale Datentyp könnte wie folgt aufgebaut sein:

Reihe/Platz	1	2	3	4
1				
2				
3				

Sofern ein Platz frei ist, wird 1 für *true* eingetragen, falls der Platz belegt ist, wird 0 für *false* eingetragen. Der Nutzer erhält nach der Überprüfung des Reservierungswunsches eine entsprechende Rückmeldung.

- b) Das Eintragen von 0 oder 1 beim Anlegen des komplexen Datentyps soll nun nicht mehr manuell vom Programmierer erfolgen, sondern mit Hilfe einer Zufallsfunktion erfolgen. Sicherlich finden Sie eine passende Methode in der Programmiersprache Ihrer Wahl, die hierfür Ihnen 0 oder 1 zurückgibt.
- c) Die Buchung des Nutzers soll nun, sofern erfolgreich, visuell hervorgehoben werden, indem bei einem nunmehr neu reservierten Platz ein „x“ eingetragen wird. Die Rückgabe könnte in diesem Fall wie folgt aussehen:

The screenshot shows a web page titled "localhost/Webgrundlagenskript/02\_Variablen/AufgabeKino/Kino.php". The content displays a success message: "Reservierung erfolgreich:" followed by a binary seating chart for a 3x5 grid. The chart is as follows:

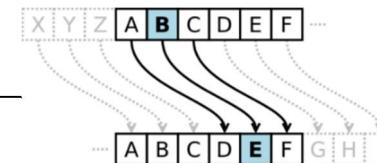
```

    _1 2 3 4 5
0: 1 1 1 0 0
1: 0 1 1 0 1
2: 0 x 0 0 1
  
```

- d) *Erweiterung für Schnelle und Kenner persistenter Datenspeicherung:* Realisieren Sie diese Aufgabe mit einer persistenten Quelle, die es ermöglicht, die Reservierung auch dauerhaft in einem Saal zu speichern. Das zufällige Ausbuchen des Saals sollte in diesem Fall natürlich wegfallen. (F)

The screenshot shows a web page titled "localhost/Webgrundlagenskript/02\_Variablen/AufgabeKino/kino.html". The content is a form with the following fields:

- A text input field labeled "Reihe" containing the value "1".
- A dropdown menu labeled "Platznummer" containing the value "2".
- Two buttons at the bottom: "Senden" and "Zurücksetzen".



## 3.7 Zusammenfassende Übungen

### 1. Einfache Cäsar-Verschlüsselung und Entschlüsselung (BA/A/DS/F)

Sie sollen ein Programm erstellen, bei der der Nutzer die Auswahl hat, eine Textnachricht zu verschlüsseln oder zu entschlüsseln. Genutzt wird die Verschlüsselung nach Cäsar. Hierbei wird ein Buchstabe um den Wert eines Schlüssels, den der Nutzer ebenfalls angeben muss, verschlüsselt bzw. bei der entsprechenden Wahl entschlüsselt. Folgende Abbildung zeigt zum Beispiel die Verschiebung um drei Stellen, der Schlüssel ist insofern 3.

Nachdem die Nutzerwahl entgegengenommen wurde, wird eine Klasse instanziert. Diese verschlüsselt durch den Aufruf einer Methode den übergebenen Text mit dem übergebenen Schlüssel und gibt diesen anschließend an den Nutzer zurück. Sollte der Nutzer hingegen eine Entschlüsselung wünschen, wird der Text anhand des übergebenen Schlüssels entschlüsselt und anschließend an den Nutzer angezeigt.

Wir vereinfachen die Aufgabenstellung dahingehend, dass nur die Übersetzung von Großbuchstaben angedacht ist. Eingegebene Kleinbuchstaben werden mittels eines passenden Befehls in Großbuchstaben von unserem Programm gewandelt. Sonderzeichen sowie Satzzeichen bleiben vorerst außer Acht. Nutzen Sie zur Lösung dieser Aufgabe die ASCII-Tabelle.

*Erweiterung für Schnelle:* Ermöglichen Sie auch die Verschlüsselung von Kleinbuchstaben sowie von Satzzeichen.

*Erweiterung für ganz Schnelle:* Befähigen Sie Ihr Programm Vorschläge zu unterbreiten, was eine mögliche korrekte Entschlüsselung einer Nachricht sein könnte. (F)

ASCII ist eine Abkürzung für (American Standard Code for Information Interchange). Dieser Standard hat sich bis heute bewährt. Dieser Standard definiert 128 Zeichen, die auf den Computern der westlichen Welt am häufigsten gebraucht werden. Dies beinhaltet beispielsweise alle Buchstaben des lateinischen Alphabets, alle Ziffern und weitere Sonder- und Steuerzeichen.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	000	<b>NUL</b> (null)	32	20	040	&#32;	<b>Space</b>	64	40	100	&#64;	<b> </b>	96	60	140	&#96;	<b>`</b>
1	1 001	041	<b>SOH</b> (start of heading)	33	21	041	&#33;	<b>!</b>	65	41	101	&#65;	<b>A</b>	97	61	141	&#97;	<b>a</b>
2	2 002	042	<b>STX</b> (start of text)	34	22	042	&#34;	<b>"</b>	66	42	102	&#66;	<b>B</b>	98	62	142	&#98;	<b>b</b>
3	3 003	043	<b>ETX</b> (end of text)	35	23	043	&#35;	<b>#</b>	67	43	103	&#67;	<b>C</b>	99	63	143	&#99;	<b>c</b>
4	4 004	044	<b>EOT</b> (end of transmission)	36	24	044	&#36;	<b>\$</b>	68	44	104	&#68;	<b>D</b>	100	64	144	&#100;	<b>d</b>
5	5 005	045	<b>ENQ</b> (enquiry)	37	25	045	&#37;	<b>%</b>	69	45	105	&#69;	<b>E</b>	101	65	145	&#101;	<b>e</b>
6	6 006	046	<b>ACK</b> (acknowledge)	38	26	046	&#38;	<b>&amp;</b>	70	46	106	&#70;	<b>F</b>	102	66	146	&#102;	<b>f</b>
7	7 007	047	<b>BEL</b> (bell)	39	27	047	&#39;	<b>!</b>	71	47	107	&#71;	<b>G</b>	103	67	147	&#103;	<b>g</b>
8	8 010	050	<b>BS</b> (backspace)	40	28	050	&#40;	<b>(</b>	72	48	110	&#72;	<b>H</b>	104	68	150	&#104;	<b>h</b>
9	9 011	051	<b>TAB</b> (horizontal tab)	41	29	051	&#41;	<b>)</b>	73	49	111	&#73;	<b>I</b>	105	69	151	&#105;	<b>i</b>
10	A 012	052	<b>LF</b> (NL line feed, new line)	42	2A	052	&#42;	<b>*</b>	74	4A	112	&#74;	<b>J</b>	106	6A	152	&#106;	<b>j</b>
11	B 013	053	<b>VT</b> (vertical tab)	43	2B	053	&#43;	<b>+</b>	75	4B	113	&#75;	<b>K</b>	107	6B	153	&#107;	<b>k</b>
12	C 014	054	<b>FF</b> (NP form feed, new page)	44	2C	054	&#44;	<b>,</b>	76	4C	114	&#76;	<b>L</b>	108	6C	154	&#108;	<b>l</b>
13	D 015	055	<b>CR</b> (carriage return)	45	2D	055	&#45;	<b>-</b>	77	4D	115	&#77;	<b>M</b>	109	6D	155	&#109;	<b>m</b>
14	E 016	056	<b>SO</b> (shift out)	46	2E	056	&#46;	<b>.</b>	78	4E	116	&#78;	<b>N</b>	110	6E	156	&#110;	<b>n</b>
15	F 017	057	<b>SI</b> (shift in)	47	2F	057	&#47;	<b>/</b>	79	4F	117	&#79;	<b>O</b>	111	6F	157	&#111;	<b>o</b>
16	10 020	060	<b>DLE</b> (data link escape)	48	30	060	&#48;	<b>0</b>	80	50	120	&#80;	<b>P</b>	112	70	160	&#112;	<b>p</b>
17	11 021	061	<b>DC1</b> (device control 1)	49	31	061	&#49;	<b>1</b>	81	51	121	&#81;	<b>Q</b>	113	71	161	&#113;	<b>q</b>
18	12 022	062	<b>DC2</b> (device control 2)	50	32	062	&#50;	<b>2</b>	82	52	122	&#82;	<b>R</b>	114	72	162	&#114;	<b>r</b>
19	13 023	063	<b>DC3</b> (device control 3)	51	33	063	&#51;	<b>3</b>	83	53	123	&#83;	<b>S</b>	115	73	163	&#115;	<b>s</b>
20	14 024	064	<b>DC4</b> (device control 4)	52	34	064	&#52;	<b>4</b>	84	54	124	&#84;	<b>T</b>	116	74	164	&#116;	<b>t</b>
21	15 025	065	<b>NAK</b> (negative acknowledge)	53	35	065	&#53;	<b>5</b>	85	55	125	&#85;	<b>U</b>	117	75	165	&#117;	<b>u</b>
22	16 026	066	<b>SYN</b> (synchronous idle)	54	36	066	&#54;	<b>6</b>	86	56	126	&#86;	<b>V</b>	118	76	166	&#118;	<b>v</b>
23	17 027	067	<b>ETB</b> (end of trans. block)	55	37	067	&#55;	<b>7</b>	87	57	127	&#87;	<b>W</b>	119	77	167	&#119;	<b>w</b>
24	18 030	070	<b>CAN</b> (cancel)	56	38	070	&#56;	<b>8</b>	88	58	130	&#88;	<b>X</b>	120	78	170	&#120;	<b>x</b>
25	19 031	071	<b>EM</b> (end of medium)	57	39	071	&#57;	<b>9</b>	89	59	131	&#89;	<b>Y</b>	121	79	171	&#121;	<b>y</b>
26	1A 032	072	<b>SUB</b> (substitute)	58	3A	072	&#58;	<b>:</b>	90	5A	132	&#90;	<b>Z</b>	122	7A	172	&#122;	<b>z</b>
27	1B 033	073	<b>ESC</b> (escape)	59	3B	073	&#59;	<b>:</b>	91	5B	133	&#91;	<b>[</b>	123	7B	173	&#123;	<b>{</b>
28	1C 034	074	<b>FS</b> (file separator)	60	3C	074	&#60;	<b>&lt;</b>	92	5C	134	&#92;	<b>\</b>	124	7C	174	&#124;	<b> </b>
29	1D 035	075	<b>GS</b> (group separator)	61	3D	075	&#61;	<b>=</b>	93	5D	135	&#93;	<b>]</b>	125	7D	175	&#125;	<b>}</b>
30	1E 036	076	<b>RS</b> (record separator)	62	3E	076	&#62;	<b>&gt;</b>	94	5E	136	&#94;	<b>^</b>	126	7E	176	&#126;	<b>~</b>
31	1F 037	077	<b>US</b> (unit separator)	63	3F	077	&#63;	<b>?</b>	95	5F	137	&#95;	<b>_</b>	127	7F	177	&#127;	<b>DEL</b>

[www.VirtualUniversity.ch](http://www.VirtualUniversity.ch)

## 2. Kochbuchrezepte (A/DS/F)

Sie sollen ein Programm erstellen, die die vorhandenen Kochrezepte auflistet und bei der entsprechenden Wahl eines Rezeptes die Details zu diesem Rezept anzeigt. Sie können hierbei folgende Beschreibung eines Rezeptes nutzen, um ein passendes Klassendiagramm hierfür abzuleiten.

*„Ein Rezept trägt einen Namen. Zudem ist die Zubereitungszeit in Minuten vermerkt. Ein Text beschreibt zudem, wie das Rezept zuzubereiten ist. Ein Rezept verfügt zudem über eine Reihe von Zutaten. Bei jeder Zutat ist vermerkt, wie der Name der Zutat ist und wie viel von dieser Zutat dem Rezept hinzuzufügen ist. Zudem verfügt das Rezept noch über die Angabe der Kalorien je Zutat. Für jedes Rezept kann zudem die Anzahl der Kalorien berechnet werden.“*

- Erstellen Sie hierzu ein passendes Klassendiagramm unter Angabe der notwendigen Attribute und Methoden.



### Marmoriertes Erdbeer-Mascarpone Eis

Rezept ohne Eismaschine, Rezept für 1,5 l



 Bild für Druck ausblenden

Zutaten für 9 Portionen:

500 g Erdbeeren

50 g Puderzucker

Eine Eisbehälterform oder eine mit Frischhaltefolie ausgelegte Auflaufform (ca. 1,5 l) bereitstellen.  
Erdbeeren waschen, gut abtropfen lassen und putzen. Die Erdbeeren und 50 g Puderzucker mit dem Zitronensaft fein pürieren. 4 EL Erdbeerpüree für das Marmor-Muster beiseite stellen.

Mascarpone und 2 EL Puderzucker mit dem Schneebesen

- Optional für Webprogrammier: Ermöglichen Sie die persistente Speicherung der Rezepte und deren Zutaten sowie das Einlesen aus der persistenten Quelle.
- Fragen Sie sich nun, wie obiges Modell ergänzt werden müssten, wenn sich die Anforderung an Ihre Software dahingehend ändern würde, dass eine Zutat nun auch mehrfach in unterschiedlichen Rezepten vorkommen könnte.

### 3.8 Reflexionsphase

Zu Beginn des Schuljahres haben Sie sich Gedanken gemacht, wie Ihre individuelle Lernentwicklung im Verlauf der Ausbildung aussehen könnte. Bitte reflektieren Sie heute auch unter Einbezug Ihrer ursprünglichen Vorstellungen aus Kapitel 1.4 folgende Aspekte.

**1. Zielerreichung:** Sind Sie gerade auf einen guten Weg, Ihre Ziele zu erreichen?

**2. Gibt es sinnvolle Anpassungen Ihres bisherigen Plans?**

**3. Zeigen** Sie mit einem **Mitschüler** Ihrer Wahl Ihre Planung und bitten Sie diesen um Feedback zu Ihren Plänen.

**4. Besprechen** Sie Ihre Ziele mit Ihrem **Lehrer**. Halten Sie die Ergebnisse beider Besprechungen, die Sie für sinnvoll erachten, im nachfolgenden Feld fest.

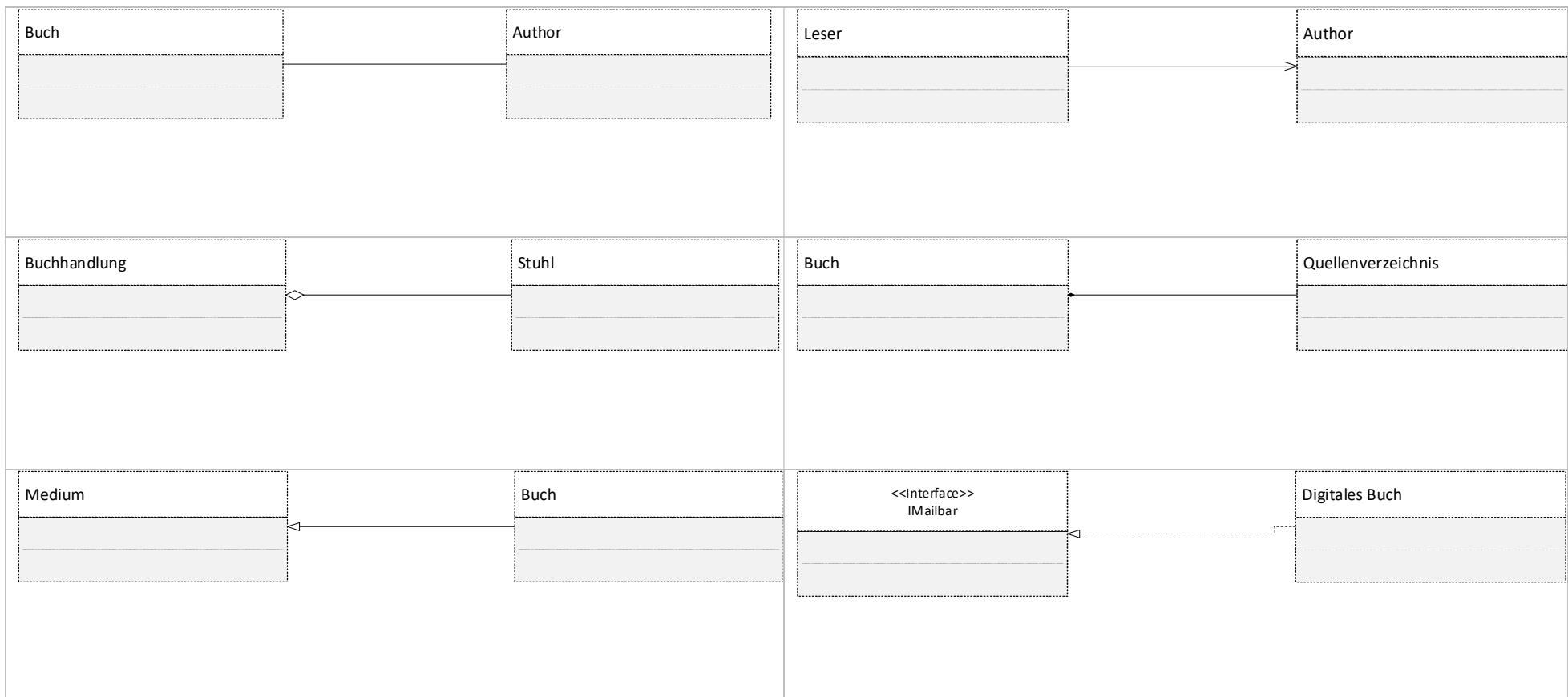


## 4. Vertiefung der objektorientierten Analyse und Design

Jetzt beginnt es langsam interessant zu werden. Sollten Sie die bisherigen Inhalte evtuelle als etwas mühsam anzueigendes Wissen wahrgenommen haben, können Sie sich nun freuen, da durch die vielfältigen Möglichkeiten der Verknüpfung von Klassen Leben in das objektorientierte Design von Problemstellungen kommt.

### 4.1 Beziehungen zwischen Klassen

Natürlich ist es auch möglich, dass Objekte miteinander kommunizieren. Dies erfolgt über den Aufruf von Methoden von einem Objekt durch ein anderes Objekt. Um eine Methode bei einem anderen Objekt aufrufen zu können, benötigt das aufrufende Objekt lediglich die Referenz auf das aufzurufende Objekt. Man spricht hierbei von einer Assoziation. Dabei gibt es verschiedene Arten von Assoziationen im Klassendiagramm:



## Übungen

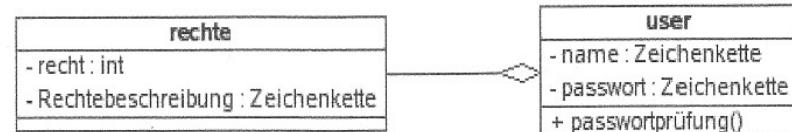
1. Prüfen Sie folgende Klassendiagramme und notieren Sie jeweils ob diese korrekt dargestellt sind. Sofern diese falsch sind, zeichnen Sie das richtige Klassendiagramm ein.

Klassendiagramm	richtig <input type="checkbox"/> falsch <input type="checkbox"/>	richtiges Klassendiagramm
<pre> classDiagram     class Person     class Student     Person &lt; -- Student   </pre>	Was ist falsch?	

Klassendiagramm	richtig <input type="checkbox"/> falsch <input type="checkbox"/>	richtiges Klassendiagramm
<pre> classDiagram     class Stuhl     class Stuhllehne     Stuhl &lt; -- Stuhllehne   </pre>	Was ist falsch?	

Klassendiagramm	richtig <input type="checkbox"/> falsch <input type="checkbox"/>	richtiges Klassendiagramm
<pre> classDiagram     class Lehrerschaft     class Lehrer     Lehrerschaft &lt; -- Lehrer   </pre>	Was ist falsch?	

2. Für eine Nutzerverwaltung wurde folgendes Diagramm entwickelt.



Welche der folgenden Aussagen zu dem Diagramm ist **nicht** zutreffend?

Tragen Sie die Ziffer vor der **nicht** zutreffenden Aussage in das Kästchen ein.

- 1 „recht“ ist eine Eigenschaft der Klasse „rechte“.
- 2 „recht“ ist nur in der eigenen Klasse sichtbar.
- 3 Die Methode „passwortprüfung()“ ist public.
- 4 Die Klasse „user“ erbt von der Klasse „rechte“.
- 5 Das Diagramm ist ein UML-Klassendiagramm.

3. Ein Server kann ein Druckserver, ein Applikationsserver oder ein Datenbankserver sein. Alle Server verfügen über das Attribut IP-Adresse sowie über die Methode, gebenIPAdresse. Ein Druckserver kennt alle ihm zugeordneten Drucker. Jeder Drucker befindet sich in einem bestimmten Raum, der jeweils eine Raumnummer sowie eine Größenangabe des Raumes hat. Ein Drucker verfügt über zwei Attribute sowie über eine Methode, die Sie jeweils frei wählen können.

## 4.2 Exkurs: Strukturierung von Programmcode

Ab einer gewissen zu erwartenden Programmkomplexität zwingt sich eine sinnvolle Strukturierung von Programmcode geradezu auf. Unter der Strukturierung des Programmcode kann im einfachsten Fall die Zuordnung von Programmcode zu Klassen und im nächsten Schritt die Zuordnung von Klassen zu Bereichen verstanden werden.

**Arbeitsauftrag:** Recherieren Sie, welche Strukturierungsmuster es in der Programmiersprache Ihrer Wahl gibt.

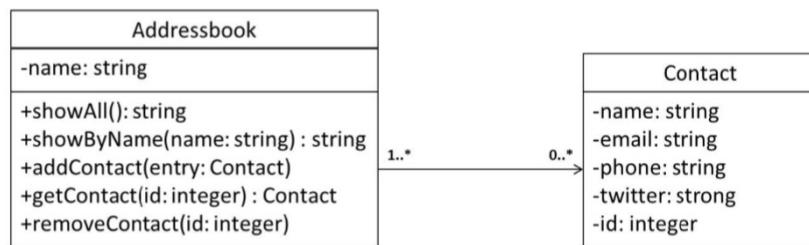


Link zu Erklärvideos: [https://klaraoppenheimer-my.sharepoint.com/:g/personal/zobel\\_klara-oppenheimer-schule\\_de/EldhPZOvkFhLpNuXX3AKPRsBw9OT-pUbmK11Vdy0LBkorw?e=DqvAab](https://klaraoppenheimer-my.sharepoint.com/:g/personal/zobel_klara-oppenheimer-schule_de/EldhPZOvkFhLpNuXX3AKPRsBw9OT-pUbmK11Vdy0LBkorw?e=DqvAab)



## Übungspool zu Beziehungen zwischen Klassen

1. **Adressbuch:** Sie sollen eine einfache Webseite erstellen, die im lokalen Netzwerk die Verwaltung von Adressen ermöglicht. Ein Kollege hat hierbei schon etwas Vorarbeit geleistet und folgendes Klassendiagramm erstellt, dass den Aufbau im Backend-Bereich der Webapplikation darstellt. (BA/A/DS)



Beachten Sie, dass die Getter- und Settermethoden von der **Contact**-Klasse im Diagramm nicht abgebildet wurden, jedoch durchaus programmiert werden müssen.

**Adressbuch: Privat**

ID	Name	Email	Phone	Twitter
1	Ken Tern	ken.tern@mail.de	+49 221 3982781	@kentern
2	Bill Iger	bill.iger@gmx.de	+49 211 9821348	@billiger
3	Flo Kati	flo.kati@web.de	+49 251 9346441	@flokati
4	Ingeborg Mirwas	inge.mirwas@post.de	+49 228 4663289	@borgmirwas
5	Ann Schweigen	ann.schweigen@gmx.de	+49 231 6740921	@annschweigen
6	Mark Enschuh	mark.enschuh@gmail.com	+49 234 4565657	@markenschuh
7	Lee Köhr	lee.koehr@mail.de	+49 561 8976761	@leekoehr
8	Pit Schnass	pit.schnass@post.de	+49 721 4545754	@pitschnass

**Adressbuch: Arbeit**

ID	Name	Email	Phone	Twitter
1	Phil Tertüte	phil.tertuete@company.de	+49 177 1786756	@philtertuete
2	Flo Kati	flo.kati@laden.com	+49 161 2336541	@bm.kati
3	Andreas Kreuz	andreas.kreuz@bazaar.de	+49 163 3442889	@asbazaar
4	Erkan Alles	erkan.alles@solver.de	+49 171 1442553	@easolver
5	Mark Reece	mark.reece@media.de	+49 151 5345612	@mrmedia
6	Roy Bär	roy.baer@media.de	+49 151 5477889	@rbmedia
7	Mario Nette	mario.nette@media.de	+49 151 5113341	@mnmedia
8	Klaus Uhr	klaus.uhr@media.de	+49 151 6743431	@kumedia

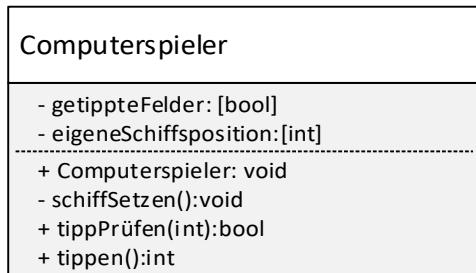
Das Programm soll eine Auswahl zwischen privates und berufliches Adressbuch anbieten. Wichtig ist außerdem, dass getätigte Änderungen auch noch nach einem erneuten Aufruf der Seite gespeichert sind.

2. **Hotelzimmerverwaltung:** Sie sollen eine HotelzimmERVERWALTUNG erstellen, die es ermöglichen soll, für Ihr fiktives Hotel freie Zimmer buchen zu können. Es soll sowohl möglich sein, sich alle freien Zimmer anzeigen, als auch ein freies Zimmer(checkIn) zu buchen. Weiterhin soll es möglich sein auszuchecken, also ein zuvor belegtes Zimmer wieder freizugeben. Selbstverständlich werden Ihre Änderungen persistent festgehalten. (DS)

Erstellen Sie hierzu zuerst ein Klassendiagramm für die notwendigen Klassen und überführen Sie nach Rücksprache mit Ihrem Lehrer das Klassendiagramm anschließend in einen objektorientierten Code.

3. **Schiffe versenken Mini:** Simulieren Sie in Form einer objektorientierten Programmierung eine Miniversion des Spiels *Schiffe versenken*. Miniversion deshalb, da das Spiel folgende Reduzierungen beinhaltet: Es gibt nur  $3 \times 3$  Felder, jeder Spieler hat drei Schiffe, die zudem jeweils nur ein Feld groß sind, es treten zwei Computerspieler gegeneinander an. (F)

Konkret soll der Nutzer ein Spiel starten können. Anschließend soll sodann das eigentliche Spiel ablaufen, dessen Ergebnis an den Nutzer angezeigt bekommt. Innerhalb des Servers soll es die Klasse *Computerspieler* geben, die folgendem Klassendiagramm entsprechen könnte:



Weiterhin soll es eine Klasse *Spielmanager* geben, die nach dem Start des Programms auf dem Server instanziert wird und anschließend solange Spielrunden durchführt, bis ein Spieler alle Treffer gelandet hat. Der glückliche Gewinner soll sodann vom Spielmanager wie bereits beschrieben auf der an den Nutzer übermittelten Resultatseite angezeigt werden. Ergänzen Sie zuerst nachfolgendes Klassendiagramm und setzen Sie anschließend das Programm entsprechend der Klassendiagramme um.



#### Erweiterungen für Schnelle:

- Für jedes versenkte Schiff soll es die gleiche Anzahl an Punkten geben. Die Punkteanzahl soll als Konstante (siehe Doku zu Konstanten) in der Klasse Schiff vermerkt sein.
- Auf der Resultatwebseite sollen nun nicht nur die Endergebnisse angezeigt werden, sondern auch die Ergebnisse jeder Runde dokumentiert sein.
- Ermöglichen Sie es, dass auch ein menschlicher Spieler gegen einen Computerspieler antreten kann.
- Ermöglichen Sie nun ein Spiel mit  $10 \times 10$  Stellen und unterschiedlichen Schiffsgrößen ( $2 \times 1, 1 \times 2, 1 \times 3, 1 \times 4, 1 \times 5$ ).

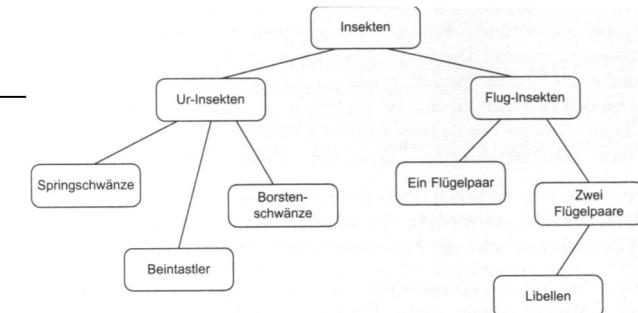


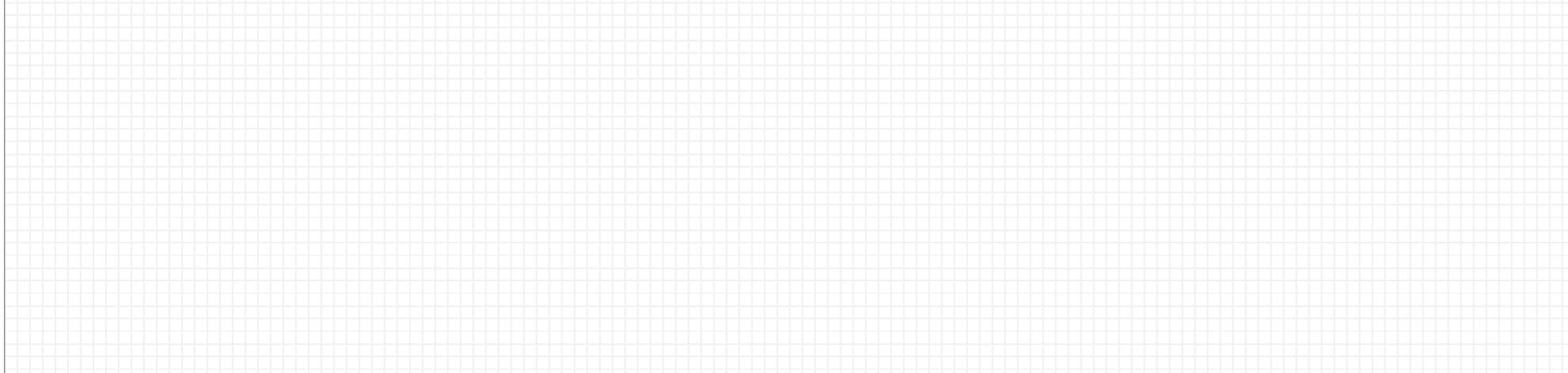
Abbildung 1: Oestreich, B., Analyse und Design mit UML 2.1, S. 49.

Das Konzept der Vererbung ist wie viele andere menschliche Konzepte aus der Natur abgeleitet.<sup>5</sup> Die Ermittlung einer hierarchischen Struktur hat zum Beispiel eine lange Tradition in der Klassifizierung der auf der Welt vorkommenden Insekten.

Hierbei werden Insekten zu Insektenfamilien und Insektenklassen geordnet. Unterhalb der Oberklasse Insekt gibt es die Unterklassen Ur-Insekt sowie Flug-Insekt. Die Gruppe der Fluginsekten wird dann weiterhin unterteilt in die Insekten mit einem Flügelpaar und Insekten mit zwei Flügelpaaren.

Wichtig ist hierbei, dass alle Insekten einer Klasse über identische Eigenschaften verfügen und die Unterklassen jeweils nur eine Spezialisierung hiervon darstellen. Die von der Oberklasse abgeleiteten Unterklassen verfügen damit automatisch über die Eigenschaften der Oberklasse, da diese an die Unterklasse vererbt werden. Daher ergibt sich auch der Name dieses Prinzips, der Vererbung. Da dieses Prinzip auch in der Objektorientierten Programmierung (OOP) eine weite Verbreitung findet, gibt es nicht nur in den Programmiersprachen entsprechende Elemente sondern auch in der allseits beliebten Notationssprache UML. Ein Pfeil ausgehend von der Unterklasse und zeigend auf die Oberklasse dient der Darstellung dieser Beziehung.

**Beispiel:** Der Buchhandel führt die Vererbung ein. Fortan soll es neben Bücher auch DVD's geben. Weitere Artikelkategorien sollen in Zukunft folgen.



<sup>5</sup> Weitere Informationen finden Sie unter dem Schlagwort Bionik.

```
C# - Medium
abstract class Medium
{
    private string isbn;
    private string title;
    private string author;
    private double price;

    public Medium(string isbn, string title, string author, double price)
    {
        this.isbn = isbn;
        this.title = title;
        this.author = author;
        this.price = price;
    }

    public string getIsbn()
    {
        return this.isbn;
    }

    public void setIsbn(string isbn)
    {
        this.isbn = isbn;
    }
}
```

```
C# - Book
class Book : Medium
{
    private int edition;

    public Book(string isbn, string title, string author, double price, int edition) : base(isbn, title, author, price)
    {
        this.edition = edition;
    }

    public int getEdition()
    {
        return this.edition;
    }

    public void setEdition(int edition)
    {
        if(edition>0)
        {
            this.edition = edition;
        }
    }
}
```

```
C# - DVD
class DVD: Medium
{
    private double size;

    public DVD(string isbn, string title, string author, double price,
              double size) : base(isbn, title, author, price)
    {
        this.size = size;
    }

    public double getSize()
    {
        return this.size;
    }

    public void setEdition(double size)
    {
        if (size > 0 && size <1000)
        {
            this.size = size;
        }
    }
}
```

```
C# - Programm
class Program
{
    static void Main(string[] args)
    {
        Book book1 = new Book("123456789", "1984", "George Orwell", 20.4, 1);
        DVD dvd1 = new DVD("123456789", "1984", "George Orwell", 20.4, 650);
    }
}
```

<pre> &lt;?php abstract class Medium {     private \$isbn;     private \$title;     private \$author;     private \$price;      function __construct(string \$isbn, string \$title, string \$author,                          float \$price)     {         \$this-&gt;isbn = \$isbn;         \$this-&gt;title = \$title;         \$this-&gt;author = \$author;         \$this-&gt;price = \$price;     }      public function getIsbn(): string     {         return \$this-&gt;isbn;     }      public function setIsbn(string \$isbn): void     {         \$this-&gt;isbn = \$isbn;     } }  class DVD extends Medium {     private \$size;      function __construct(string \$isbn, string \$title, string \$author, float     \$price, float \$size)     {         \$this-&gt;size = \$size;         parent::__construct(\$isbn, \$title, \$author, \$price);     }      public function getSize(): float     {         return \$this-&gt;size;     }      public function setEdition(float \$size): void     {         if (\$size &gt; 0 &amp;&amp; \$size &lt; 1000)         {             \$this-&gt;size = \$size;         }     } } </pre>
---

<pre> class Book extends Medium {     private \$edition;      function __construct(string \$isbn, string \$title, string \$author, float \$     price, int \$edition)     {         \$this-&gt;edition = \$edition;         parent::__construct(\$isbn, \$title, \$author, \$price);     }      public function getEdition(): int     {         return \$this-&gt;edition;     }      public function setEdition(int \$edition): void     {         if (\$edition &gt; 0)         {             \$this-&gt;edition = \$edition;         }     } } </pre>
--

<pre> \$book1 = new Book("123456789", "1984", "George Orwell", 20.4, 1);  \$dvd1 = new DVD("123456789", "1984", "George Orwell", 20.4, 650); </pre>
---

```
import abc

class Medium(metaclass=abc.ABCMeta):

    def __init__(self, isbn, title, author, price):
        self.isbn = isbn
        self.title = title
        self.author = author
        self.price = price

    def getIsbn(self):
        return self.isbn

    def setIsbn(self, isbn):
        self.isbn = isbn

#Hinweis: Eine abstrakte Klasse benötigt in Python noch eine abstrakte Methode, um eine
#implementierung dieser zu verhindern.
```

Python - Medium

```
class Book (Medium):

    def __init__(self, isbn, title, author, price, edition):
        self.edition = edition
        super().__init__(isbn, title, author, price)

    def getEdition(self):
        return self.edition

    def setEdition(self, edition):
        if (edition > 0):
            self.edition = edition
```

Python - Book

```
class DVD (Medium):

    def __init__(self, isbn, title, author, price, size):
        self.size = size
        super().__init__(isbn, title, author, price)

    def getSize(self):
        return self.size

    def setEdition(self, size):
        if (size > 0 and size < 1000):
            self.size = size
```

Python - DVD

```
book1 = Book("123456789", "1984", "George Orwell", 20.4, 1)
dvd1 = DVD("123456789", "1984", "George Orwell", 20.4, 650)
```

Python - Programm

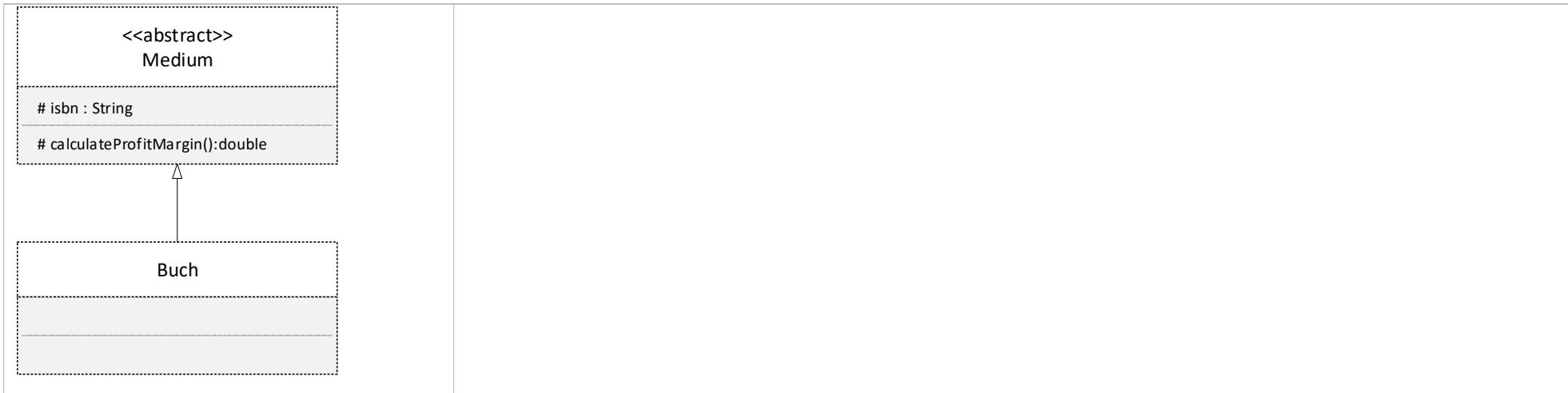
**Arbeitsauftrag:** Vergleichen Sie nun die verschiedenen Programmiersprachen zueinander. Notieren Sie **Unterschiede** und **Gemeinsamkeiten**.



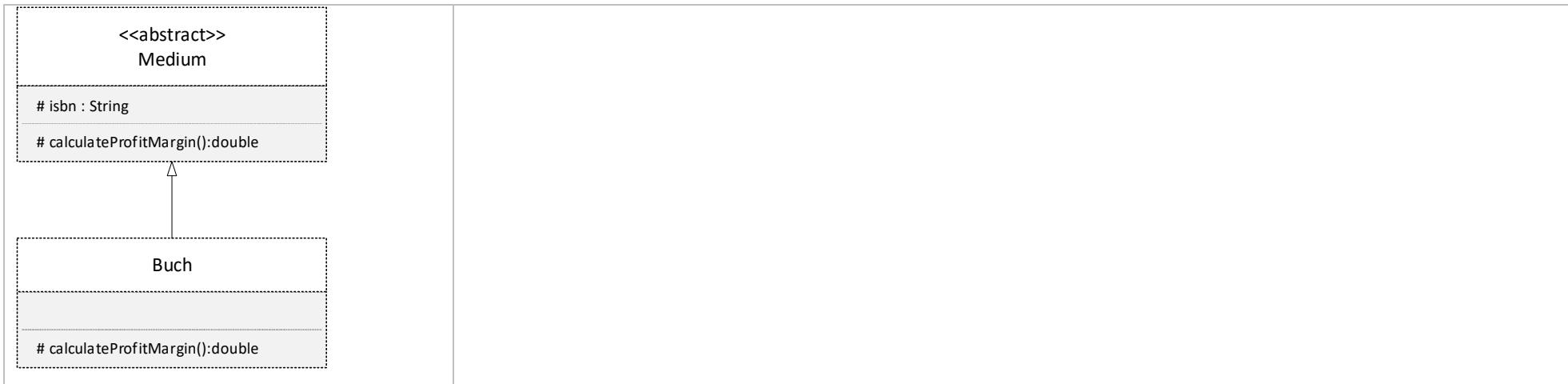
Sprachenvergleich

#### 4.3.2 Methoden in der Vererbungsstruktur

##### Methoden der Oberklasse nutzen



##### Methoden der Oberklasse überschrieben



### Implementieren von abstrakten Methoden der Oberklasse

<pre>&lt;&lt;abstract&gt;&gt; Medium  + printCompleteTitle(): String {abstract}</pre>	
<pre>Buch  + printCompleteTitle(): String</pre>	

### Fragen

**F:** Wie lang ist bei PHP eigentlich die Lebensdauer der instanziierten Objekte?

**A:** Also die Lebensdauer von Objekten endet auch bei PHP mit dem Ende der Programmausführung.

Zu beachten ist allerdings, dass bei Desktopapplikationen, die zum Beispiel in C# oder Java programmiert werden, die Kommunikation zwischen dem Nutzer und den Programm über die Konsole oder ein Interaktionsfenster läuft und die Kommunikation noch Teil der Programmausführung ist.

Im Webumfeld, also bei Client-Serverapplikationen, ist es jedoch so, dass die Ausführung von PHP und damit auch verbunden die Lebensdauer eines Objektes mit dem Senden der Http-Response Nachricht vom Server abgeschlossen ist. Zustandsänderungen von Objekten oder auch deren Neuererzeugung oder Löschung gehen damit verloren, es sei denn, die Objekte werden in einer persistenten Quelle gespeichert. Eine Vertiefung zur Lebenszeit von Objekten bei PHP finden Sie mit folgendem Link: <https://www.tutorials.de/threads/oop-lebensdauer-von-objekten.22258/>



Link zu Erklärvideos: [https://klaraoppenheimer-my.sharepoint.com/:f/g/personal/zobel\\_klara-oppenheimer-schule\\_de/EhmPSXxILwBOt4AukF2eo7oBM-qssS8bi9em8MfhP1f14g?e=X1iBRK](https://klaraoppenheimer-my.sharepoint.com/:f/g/personal/zobel_klara-oppenheimer-schule_de/EhmPSXxILwBOt4AukF2eo7oBM-qssS8bi9em8MfhP1f14g?e=X1iBRK)

**F:** Kann man eigentlich verhindern, dass eine erbende Klasse eine Methode überschreibt?

**A:** Dies ist möglich. In Java z.B. durch die Angabe des zusätzlichen Modifizierers final an der Methodendeklaration

**Arbeitsauftrag:** Klären Sie, wie dies in der Sprache Ihrer Wahl ist und notieren Sie im folgenden Feld das Ergebnis:

**A:** \_\_\_\_\_



## Übungspool zur Vererbung

- 1. Fahrzeugverwaltung:** Im nachfolgenden Code haben sich leider und völlig überraschend ein paar Fehler eingeschlichen. Zum einen müssen Sie die Fehler im Code finden und korrigieren, zum anderen fehlende Elemente, die zwingend erforderlich sind, um eine Funktionsfähigkeit des Codes zu bewirken, ergänzen. (BA/A)

```
<?php
class Fahrzeug
{
    private $geschwindigkeit = 0;
    public $hupe="Tutttut";

    function beschleunigen($wert)
    {
        $this->geschwindigkeit += $wert;
    }

    function hupen()
    {
        echo $this->hupe;
    }

    function ausgabe()
    {
        echo "Geschwindigkeit: $this->geschwindigkeit<br />";
    }
}

PKW extends Fahrzeug
{
    private insassen = 0;

    function einsteigen($anzahl)
    {
        $this->insassen += $anzahl;
    }

    function aussteigen($anzahl)
    {
        $this->insassen -= $anzahl;
    }

    function ausgabe()
    {
        echo "Insassen: $this->insassen "."<br>";
        parent::ausgabe();
    }
}
```

```
class Zweirad
{
    private $heizgriffe;
    private $motorLaeuft=false;

    function zweiradAnkicken( )
    {
        if($staerke>20)
        {
            if($this->motorLaeuft= false)
            {
                $this->motorLaeuft=true;
                echo "Huihhh - Motor läuft."<br>;
            }
            else
            {
                echo "Krrr- Motor lief bereits."<br>;
                echo "Antreten eines Zweirads kann bei ". $this->geschwindigkeit . " kmh gefährlich sein!";
            }
        }
    }

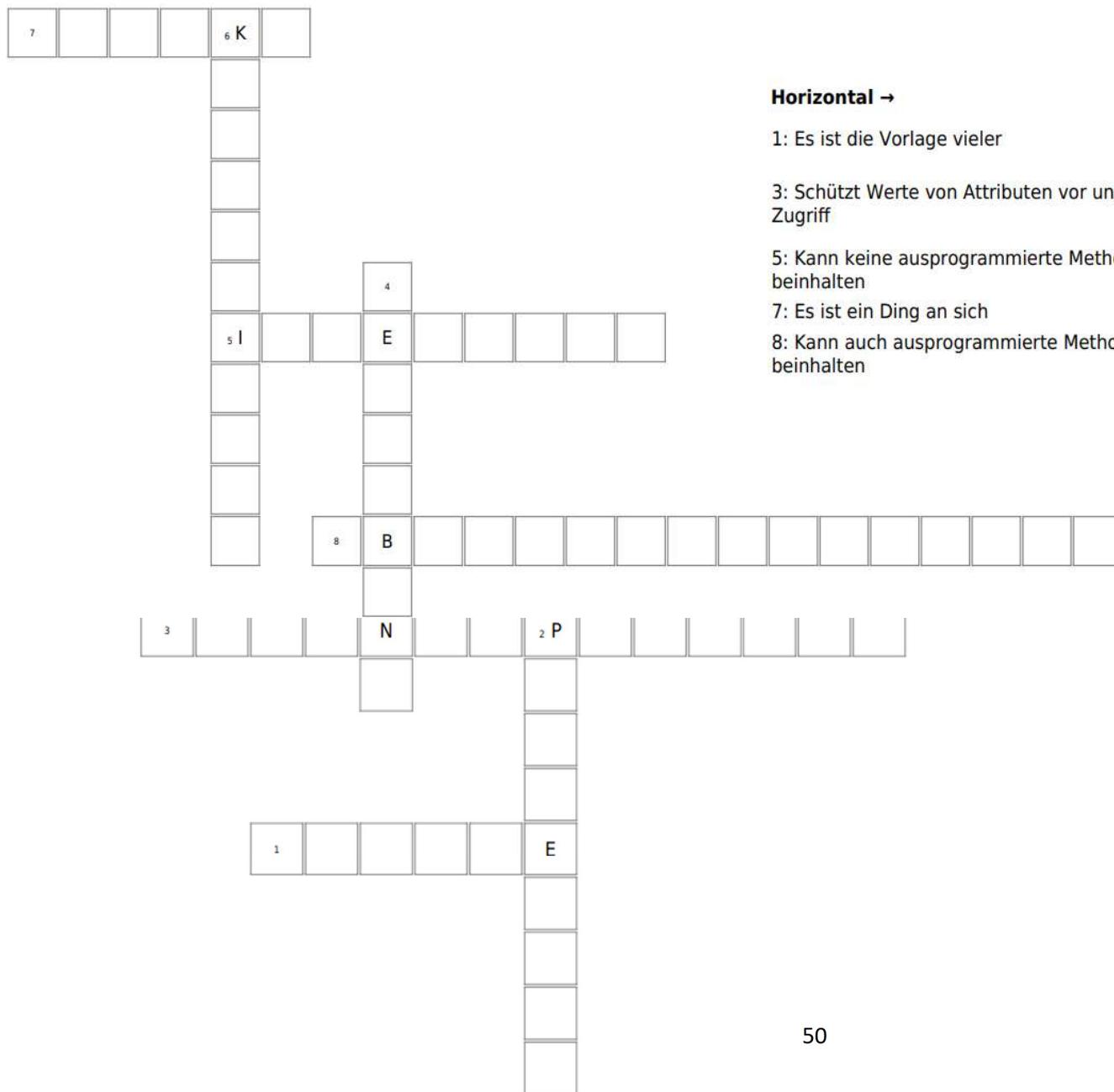
    function ausgabe()
    {
        echo "Heizgriff vorhanden: $this->heizgriffe ";
        parent::ausgabe();
    }
}
$fiat = new PKW ;
$fiat->ausgabe();

$fiat->einstiegen(3);
$fiat->beschleunigen(30);
$fiat->ausgabe();

$fiat->beschleunigen(-30);
$fiat->ausgabe();

$fiat->aussteigen(1);
$fiat->ausgabe();
$bmw = Zweirad();
$bmw->ausgabe();
$bmw->beschleunigen(30);
$bmw->ausgabe();
$bmw->hupen();
$bmw->zweiradAnkicken(60);
$bmw->zweiradAnkicken(60);
```

## 2. Kreuzworträtsel zur Objektorientierte Programmierung (BA/A)



## Horizontal →

- 1: Es ist die Vorlage vieler
  - 3: Schützt Werte von Attributen vor unberechtigtem Zugriff
  - 5: Kann keine ausprogrammierte Methoden beinhalten
  - 7: Es ist ein Ding an sich
  - 8: Kann auch ausprogrammierte Methoden beinhalten

Vertikal S

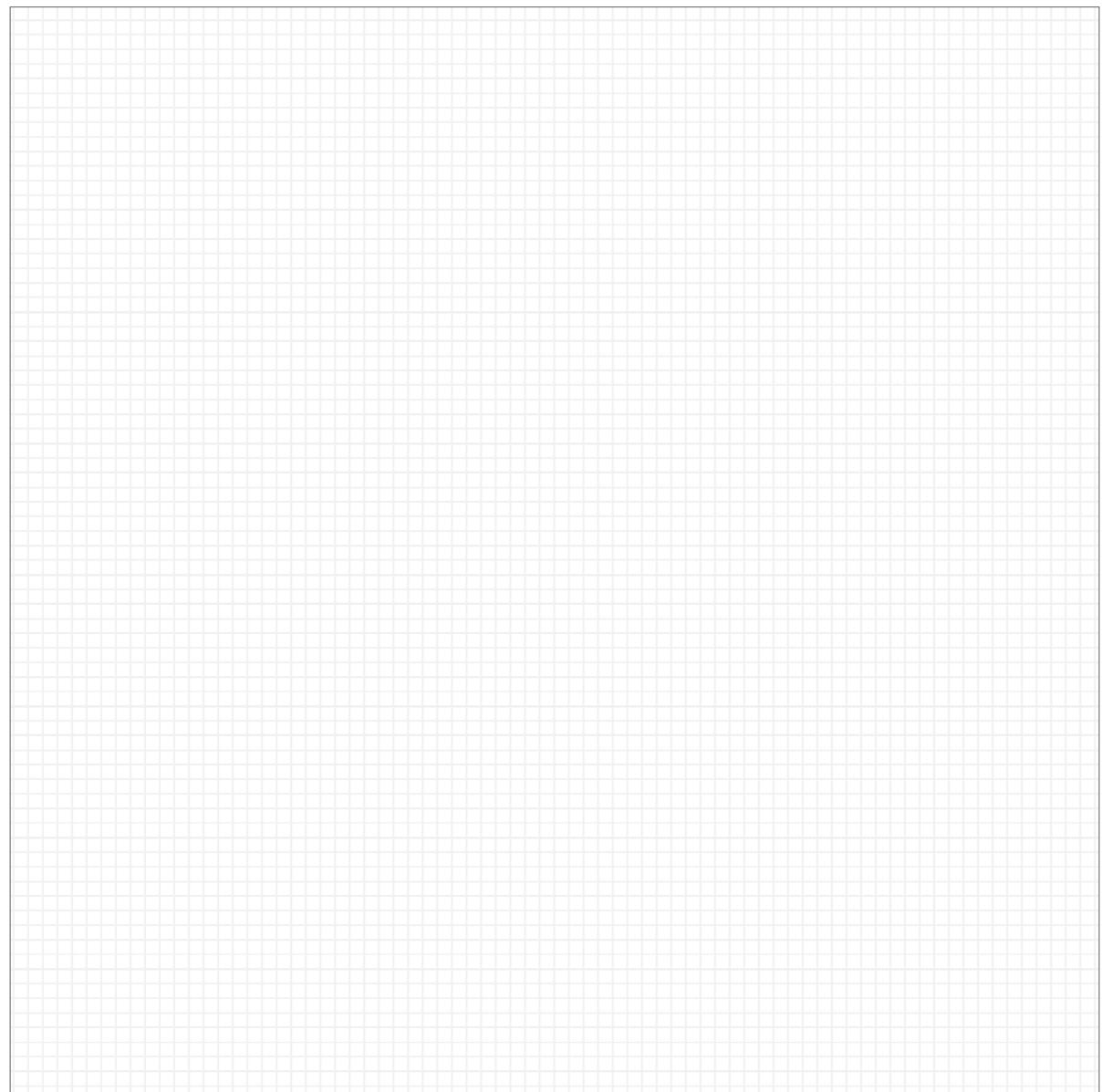
- 2: Sichtbarkeit, die auch in der Unterklasse verfügbar ist
  - 4: Das Prinzip ermöglicht das Zusammenfassen gleicher Zusammenhänge in einer abstrakten Ebene
  - 6: Besteht aus einem Ganzen und einen abhängigen Teil

### 3. Raumverwaltung (DS/F)

Ihre Aufgabe ist es, eine objektorientierte Datenstruktur zur Verwaltung von Räumen an einer Berufsschule zu entwerfen.

- Jeder Raum wird durch seine Nummer gekennzeichnet.
- Ein Seminarraum ist ein Raum, der sich durch die Fläche seiner Tafel auszeichnet. Außerdem ist für einen Seminarraum relevant, ob dort ein Beamer vorhanden ist.
- Ein Klassenzimmer ist ein Raum, für den neben der Fläche der Tafel auch die Anzahl der Fenster speichert.
- Neben den Räumen mit Tafel gibt es auch Räume, die als Büros genutzt werden. Für diese Räume ist ein wichtiges Attribut, auf welcher Etage sie sich befinden.
- Jeder Raum verfügt über die Methode *lichtAnschalten*
- Ein Lehrerbüro ist ein Büro, für das der Name des Lehrers wichtig ist.
- Ein Rechnerraum ist ein Büro, das mit einer bestimmten Datenübertragungsrate an das Netzwerk der Berufsschule angeschlossen ist.
- Alle Räume mit Tafeln sind entweder Seminarräume oder Klassenzimmer. Für diese Räume ist bekannt, wie viele Sitzplätze sie haben. Es kann aber Büros geben, die weder Lehrerbüros noch Rechnerräume sind.

Jedes Büro verfügt über eine Methode, mit der die Klimaanlage angeschaltet werden kann, wobei bei Lehrerbüros diese Methode überschrieben wurde.



#### 4.3.4 Diamondproblem

**Arbeitsauftrag:** Erstellen Sie für folgenden Sachverhalt ein UML-Klassendiagramm und setzen Sie es, nachdem Sie es kurz mit Ihrem Lehrer im Leisen besprochen haben, in der Programmiersprache Ihrer Wahl um.

**Kontext:** Es gibt sowohl Landfahrzeuge als auch Wasserfahrzeuge. Beide dieser Fahrzeugtypen erben von der Klasse Fahrzeug. Diese Klasse besitzt eine abstrakte Methode `fortbewegen` sowie das Attribut `anzahlSitzplaetze` sowie `hoechstgeschwindigkeit`. Die maximale höchstgeschwindigkeit varriert. Während ein Landfahrzeug maximal 180 km/h fahren darf, ist dies bei einem Wasserfahrzeug nur 60 km/h. Weiterhin gibt es Amphibienfahrzeuge. Ein Amphibienfahrzeug erbt sowohl die Eigenschaften eines Land- als auch die eines Wasserfahrzeuges.



Link zu Erklärvideos:

## 4.5 Interface

Interfaces sind ein Element der OOP, welches zwar Methoden definiert, diese aber nicht implementiert. D.h. sie definieren Methoden mit Argumenten und evtl. Rückgabewerten, implementieren die Methode aber nicht.

C#	<pre>interface IPrintable {     void print(); }  class Book : Medium, IPrintable {     ...     public void print()     {         Console.WriteLine("I print");     } }</pre>	PHP	<pre>&lt;?php  interface IPrintable {     public function print() : void; }  class Book : Medium implements IPrintable {     ...     public function print() : void     {         echo "I print";     } }</pre>
----	--	-----	---

Python		
--------	--	--

**Arbeitsauftrag:** Vergleichen Sie nun die verschiedenen Programmiersprachen zueinander. Notieren Sie **Unterschiede** und **Gemeinsamkeiten**.



### Sprachenvergleich

## Fragen

**F:** Welchen Unterschied gibt es zwischen abstrakten Klassen und Interfaces?

**A:** Interfaces unterscheiden sich von abstrakten Klassen, weil diese sowohl eigene Methoden implementieren können als auch abstrakte Methoden vorgeben können, die die erbenden Klassen implementieren müssen. Interfaces sollen hingegen nur darstellen, was eine Klasse tun kann, aber nicht, wie es dieses Verhalten umsetzt wird.

**F:** Warum wird bei Interfaces häufig auch der Vergleich zu einem „Vertrag“ gezogen?

**A:** Ein Vertrag gibt vor, welche Regeln einzuhalten sind. Dies entspricht bei einem Interface den Methodensignaturen, die von der implementierenden Klasse einzuhalten sind. Zudem kann sich ein Dritter darauf verlassen, dass - wenn eine Klasse ein Interface implementiert - diese eben die im Vertrag definierten Methoden auch implementiert haben muss.

## Zusammenfassender Übungspool zur Vererbung, Interfaces und Abstrakten Klassen

### 1. Telefonarten (BA/AD/DS)

- a) In dieser Aufgabe betrachten wir verschiedene Arten von Telefonen. Es geht hierbei um den Entwurf einer entsprechenden Klassenhierarchie, mit der die verschiedenen Arten von Telefonen und ihren Komponenten sinnvoll in einem Programm gehandhabt werden können.
- Festnetztelefone sind Telefone, die nur an einem bestimmten Standort genutzt werden können, da sie mit dem Festnetz verbunden sind. Für diese Telefone ist der Standort bekannt.
  - Ein Akku speichert seinen Füllstand in Prozent.
  - Bei allen Telefonen wird die Lautstärke ganzzahlig in Dezibel gespeichert.
  - Ein Teil der Festnetztelefone ist stationär, d.h., sie sind fest an einem Standort, wie z.B. der Wohnung oder dem Büro, installiert und können nicht mitgenommen werden.
  - Als Gegensatz zu Festnetztelefonen gibt es Mobiltelefone. Jedes Mobiltelefon benutzt einen Akku als Komponente und die Anzahl der Tasten ist bekannt.
  - Es gibt auch Mobiltelefone mit Touchscreen. Bei diesen kann man die Eingaben per Touchscreen und per Tasten vornehmen. Für diese Telefone ist die Diagonale des Touchscreens bekannt.
  - Die Festnetztelefone, die nicht stationär sind, sind drahtlose Telefone. Solche Telefone können in einem bestimmten Radius um den Standort herum genutzt werden und benutzen einen Akku. Die Reichweite in Metern soll gespeichert werden.
  - Die Akku sind aufladbar und stellen die Methode `getLadestand()` bereit, mit der der Ladestand (0-100) des Akkus abgefragt werden kann. Außerdem haben sie eine Methode `aufladen()`, die den Akku auflädt und zurückgibt, ob das Aufladen erfolgreich war.
  - Ein öffentliches Telefon ist ein stationäres Telefon, das jeder benutzen kann. Für dieses Telefon sind die Kosten pro Zeiteinheit in Cent bekannt.
- b) Ein Kollege hat Sie darauf hingewiesen, dass es in Zukunft möglich sein soll, unterschiedliche Akkus(Nickel-Kadmium-Akku, Nickel -Metallhydrid-Akkus, Lithium-Ion-Akkus) in den Telefonen mit Akkus einzubauen zu können. Weitere Akkuarten werden in der Zukunft sicherlich noch dazu kommen. Jede Akkuart implementiert die Methode `aufladen` hierbei unterschiedlich. Wichtig ist ihrem Kollegen, dass die Schaffung neuer Akkus keine Änderung im Code der Telefone mit Akku bedarf. Er nannte hierbei das Stichwort „*Open-Closed-Prinzip*“. Er meinte dies wäre recht *solid!*“?

### 2. Kurierdienst (A/DS)

Ein Kunde bittet Sie, für folgenden Sachverhalt ein Programm zu schreiben:<sup>6</sup>

*„Ein Fahrtenbuch soll Fahrten verwalten können, die die Kurierfahrer mit den verschiedenen Fahrzeugen des Unternehmens zurücklegen. Jede Fahrt ist mit einem Fahrzeug assoziiert und speichert die zurückgelegten Kilometer einer Fahrt. Das Unternehmen verfügt als Fahrzeuge über Fahrräder, Motorroller und Kleintransporter. Diese Fahrzeugtypen verfügen wiederum über verschiedene Sätze pro gefahrenem Kilometer, mit denen die Transportentgelte ermittelt werden. Fahrrad: 1,00 EUR/km, Motorroller 2,00 EUR/km und Kleintransporter 5,50 EUR/km. Um den Umsatz bestimmen zu können, soll das Fahrtenbuch über eine entsprechende Operation zur Berechnung verfügen.“*

---

<sup>6</sup> Vgl. Programmierentrainieren, Lolacono/Wiefling/Schneider, 2018.

3. **Zooverwaltung:** Sie sollen die Software für die Kasse an einer Zooverwaltung programmieren. Die Besucher werden als Erwachsene, Kinder oder als Familien verbucht. Während der Eintritt für Kinder 3 Euro kostet, zahlt jeder Erwachsener 5 Euro und eine Familie zahlt 10 Euro pauschal. Bei einer Familie wird immer festgehalten, wieviele Erwachsenen und Kinder darin enthalten sind. Das Programm bietet dem Benutzer der Kasse einerseits die Möglichkeit, die Eintrittskarten zu verbuchen und andererseits eine Statistik abrufen zu können, wie viele Besucher der Zoo an diesem Tag hatte. Selbstverständlich werden alle Informationen in passenden Objekten gehalten. (DS)
4. **Fuhrpark:** Sie sollen für ein Autofahrtsimulationsprogramm einen passenden Fuhrpark in objektorientierter Form entwerfen. Erstellen Sie zu jeder Teilaufgabe zuerst ein passendes UML-Diagramm und setzen Sie dieses Diagramm anschließend in PHP um. Erstellen Sie zu Testzwecken immer auch im Hauptprogramm entsprechende Testinstanzen der jeweiligen Autoklassen und rufen Sie zudem die jeweils vorhandenen Funktionen fahren auf. Ihr Fuhrpark soll über folgende Autos verfügen: (DS/F)

	<b>Firebird:</b> Es handelt sich hierbei um einen Firebird. Beim Firebird ist wie bei allen anderen Autos auch der Hersteller gespeichert. Eine Besonderheit ist beim Firebird, das darüberhinaus auch die Wattzahl der Unterbodenbelichtung bekannt ist. Der Firebird gibt beim Aufruf der Methode <code>fahren „Brummbrumm“</code> aus. Der Firebird ist der Kategorie Rennwagen zugeordnet.
	<b>Weitere Rennwagen:</b> Ferner existieren zwei weitere Rennwagen in Ihrem Fuhrpark. Im Rahmen von ausgiebigen Testphasen zeigte sich, dass das obere Auto, ein Model <i>Majorette Turbo</i> , beim Aufruf der Methode <code>fahren</code> , nicht <code>„Brummbrumm“</code> macht sondern <code>„Uuhweemmnh“</code> . Zudem ist beim <i>Majorette Turbo</i> die jeweilige Anzahl der Flügeltüren gespeichert. Das rechte Auto, vom Typ <i>Fire Turbo</i> , gibt ebenfalls wie der Firebird <code>„Brummbrumm“</code> aus. Zudem ist gespeichert, ob der Turbo beim Fire Turbo angeschalten ist.
	<b>Freizeitwagen:</b> Der Fuhrpark verfügt weiterhin über einen <i>Mercedes Benz</i> vom Typ <i>AMG</i> . Dieses Auto entspricht nicht der Kategorie Rennwagen, sondern der Kategorie Freizeitwagen. Die Freizeitwagen zeichnen sich unter anderen dadurch aus, dass diese zum Teil über eine integrierte Minibar verfügen. Ob dies der Fall ist, ist in der entsprechenden Klasse vermerkt. Beim <i>Mercedes Benz</i> Typ <i>AMG</i> ist zudem vermerkt, ob eine Wegfahrsperrre vorhanden ist. Alle Freizeitwagen geben beim Aufruf ihrer Methode <code>fahren</code> ein <code>„scheeechuiiehh“</code> aus.
	<b>Beachcar:</b> Ein weiterer Freizeitwagen ist ein Beachcar. Was dieses Auto besonders macht ist die Fähigkeit, auch Sprünge vollführen zu können - in der Ausgabe erscheint beim Aufruf dieser Methode <code>JumpJump</code> . Ferner verfügt das Auto in vielen Fällen über einen ausklappbaren Sonnenschutz.
	<b>Lowrider:</b> Da die Sprungfunktionalität bei unseren Kunden besonders gut ankommt, soll es zudem vom Firebird eine Spezialisierung geben, die die Sprungfähigkeit inne hat. Dort jedoch nicht in Form der Ausgabe von <code>„JumpJump“</code> , sondern durch die Ausgabe von <code>„I am a Lowrider“</code> . Der Auftraggeber möchte dabei sichergestellt haben, dass alle Autos, die eine Sprungfunktionalität versprechen, diese auch tatsächlich ausführen können.
	<b>Fuhrparkverwalter:</b> Es soll zudem eine Klasse <i>Fuhrparkverwalter</i> geben, die über eine typischere Collection verfügt, mittels derer auf alle vorhandenen Instanzen des Fuhrparks zugegriffen werden kann. Der Fuhrparkverwalter soll zudem, nach einem Aufruf seiner Methode <code>checkFuhrpark</code> , alle Autos Probe <code>fahren</code> lassen. Der Fuhrparkverwalter ist es auch, der jedem Auto bei der Erstellung eine AutoID zuweist, die jedes Auto im Fuhrpark eindeutig identifizierbar macht.

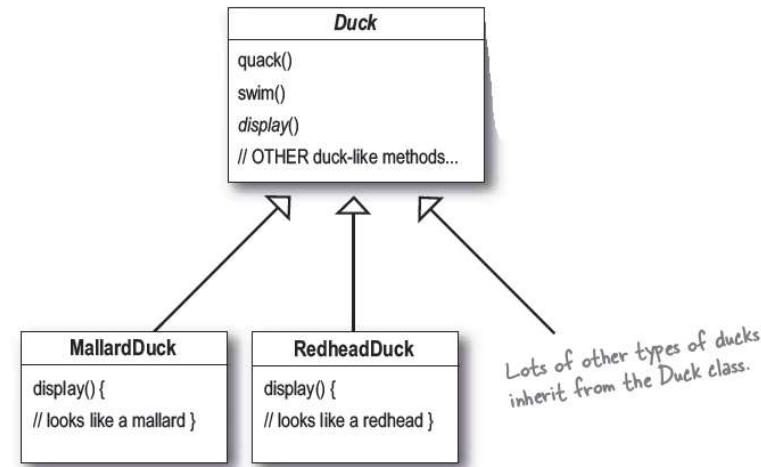
A large rectangular grid with a light gray background and a faint grid pattern, intended for drawing or writing.

- a) Erstellen Sie nun basierend auf obigen Klassendiagramm ein Programm.
  - b) Ein einzigartiger Fuhrparkverwalter: Der Kunde fordert nun im Rahmen der Programmgestaltung, dass zu jedem Zeitpunkt immer nur eine Instanz der Klasse Fuhrparkverwalter geben darf. Ein Kollege hat darauf hin von einer möglichen Umsetzung im Rahmen des sogenannten Singleton-Patterns gesprochen. Finden Sie hierzu passende Informationen im Internet .
  - c) Ein Kollege meint, wenn der Fuhrparkverwalter eine generische Liste des Typs Auto hält, würde damit eine dynamische Polymorphie ermöglicht. Versuchen Sie zu klären, was damit gemeint ist.
  - d) Freiwillige Zusatzaufgabe für Schnelle: Der Fuhrparkverwalter will springen: Ihr Fuhrparkverwalter möchte nun alle Autos seines Fuhrparks springen lassen, sofern die jeweilige Autos über eine passende Sprungfunktion verfügen. Erstellen Sie zu diesem Zweck eine Methode der Klasse Fuhrparkverwalter, die diesem Wunsch gerecht wird. Rufen Sie anschließend diese Methode auch aus dem Hauptprogramm aus auf.
5. Implementieren Sie für ein möglichst sinnvolles objektorientiertes Beispiel den sogenannten **Dijkstra Algorithmus**. Sprechen Sie Ihr Konzept vor Umsetzung bitte mir Ihren Lehrer ab.  
(DSKMNMLT)

## 5. Entwurfsmuster

### 5.1 Strategy Pattern

Grundgerüst einer Entensimulation<sup>7</sup>, das zeigt, wie verschiedene Entenarten herumschwimmen und Quak-Geräusche von sich geben. Beschreiben Sie die Abbildung stichpunktartig.

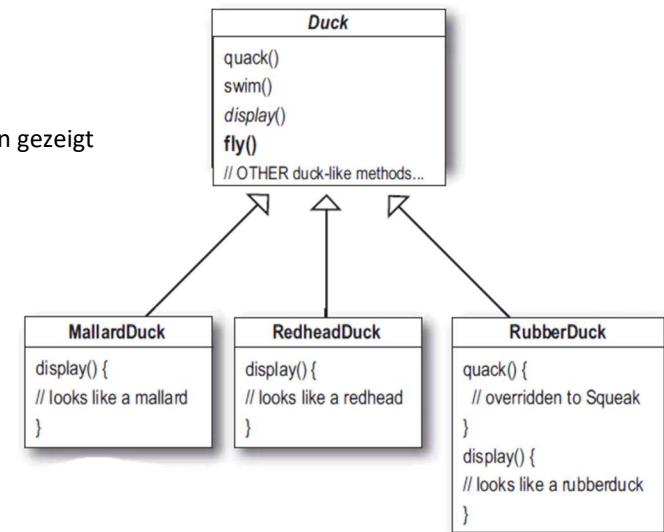


Die Unternehmensleitung hat beschlossen, dass es für die Simulation der Reiser wäre, wenn auch das Flugverhalten aller Enten gezeigt werden könnte. Was ist am folgenden Vorschlag, der nun auch zusätzlich eine RubberDuck (Gummiente) zeigt, auszusetzen?

Wäre folgende Erweiterung der RubberDuck-Klasse eine gute Lösung?

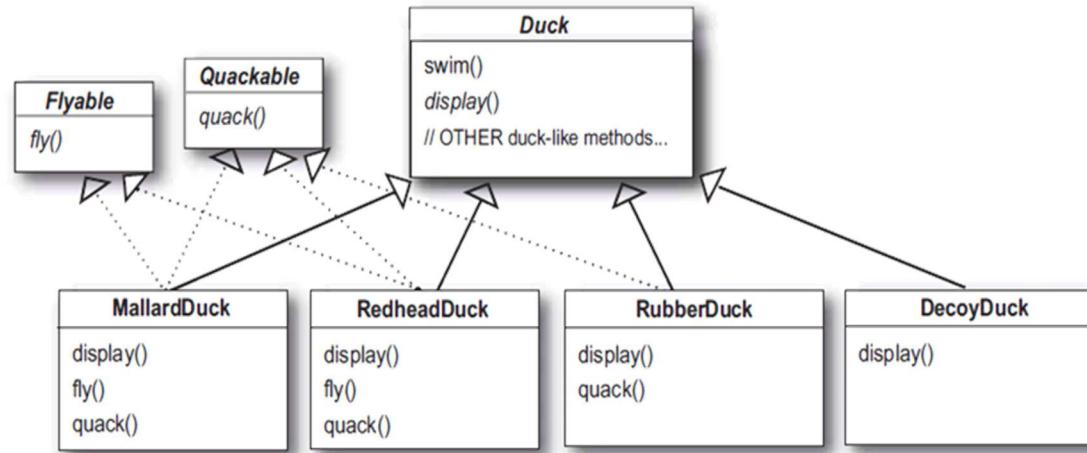
```

class RubberDuck {
    quack() { // squeak}
    display() { // rubber duck }
    fly() {
        // override to do nothing
    }
}
  
```



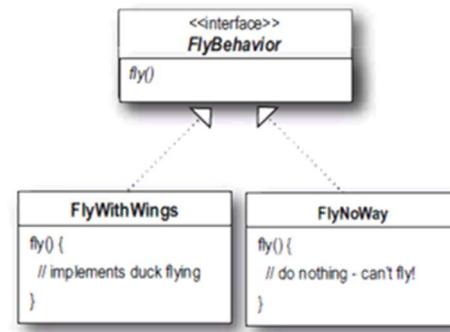
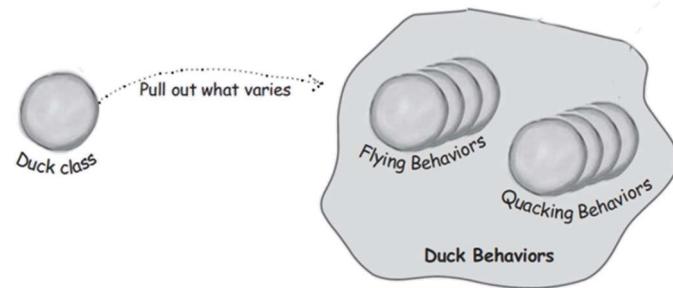
<sup>7</sup> Vgl. Freeman, Entwurfsmuster von Kopf bis Fuß, S. 2-23.

Die Unternehmensleitung möchte nun alle sechs Monate neue Enten im Programm einbauen. Dies bedeutet, dass es die bisherige Implementierung notwendig macht, bei jeder neuen Ente, das Flug- sowie das Quakverhalten in der entsprechenden Methode zu überprüfen.



**Arbeitsauftrag:** Überlegen Sie sich mit Ihrem Banknachbarn ob, Interfaces eine sinnvolle Lösung für das Problem sind. Finden Sie Vor- und Nachteile.

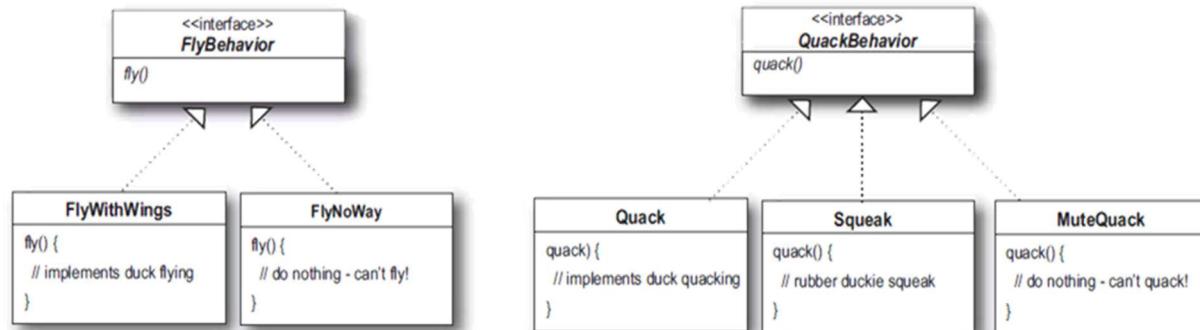
**Lösungsansatz:** Das Veränderliche herausziehen



E

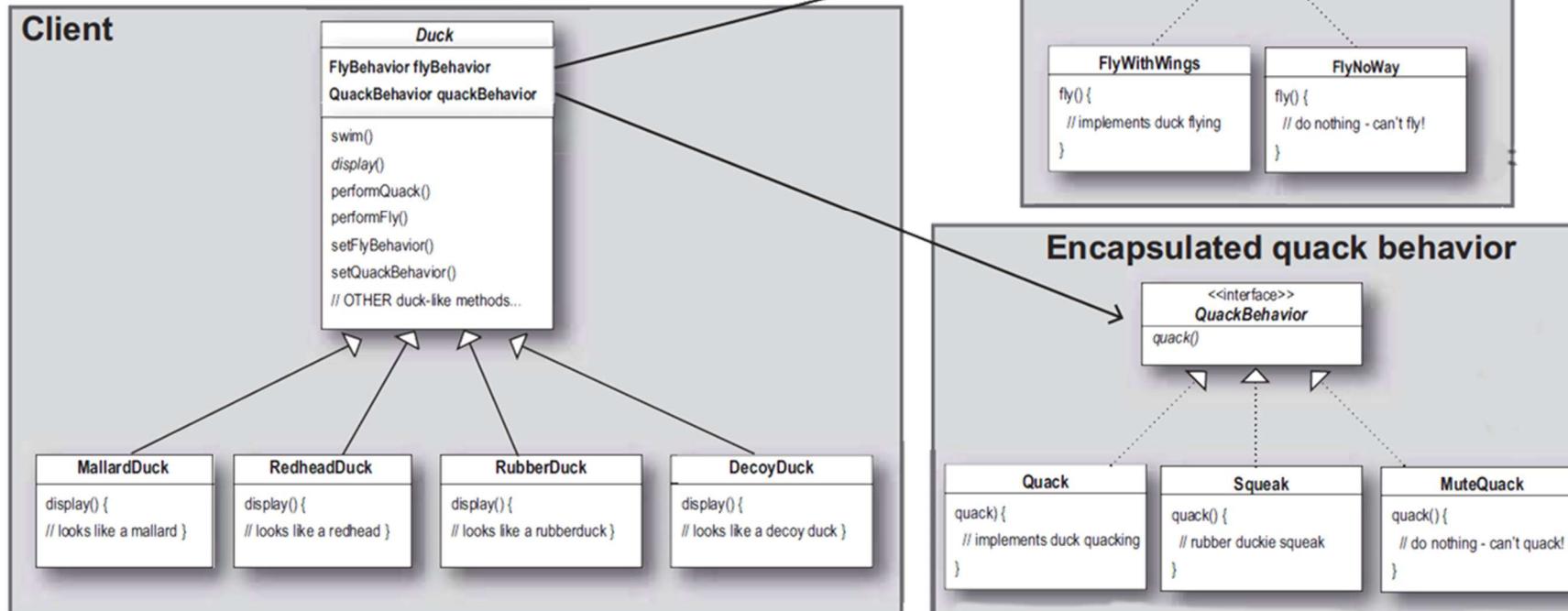
**Entwurfsprinzip:** Programmiere auf eine Schnittstelle, nicht auf eine Implementierung

Veränderliches Verhalten wird in eigene Klassen ausgelagert



-> Vorteile der Wiederverwendung nutzen ohne Ballast, den die Vererbung bringt.

Dies führt zum **Strategy-Pattern** in der SimEnte-Anwendung



**Arbeitsauftrag:** Erstellen Sie stichpunktartige Notizen, die es Ihnen ermögliche, das Pattern auch später noch nachzuvollziehen sowie in weiteren, ähnlichen Kontexten anwenden zu können.

**Arbeitsauftrag:** Setzen Sie nun das Pattern in der Sprache Ihrer Wahl um. Erzeugen Sie nach Fertigstellung der Klassenstruktur eine Ente, lassen Sie diese quaken und fliegen und weisen Sie dieser während der Laufzeit ein neues Flugverhalten zu. Testen Sie sodann die Wirksamkeit der dynamischen Polymorphie.



Link zu Erklärvideos: [https://klaraoppenheimer-my.sharepoint.com/:f/g/personal/zobel\\_klara-oppenheimer-schule\\_de/EloJcg3VjD5Jj6SH8wqRpCoBoDRPBAFW\\_CZldWGxUlDkRw?e=rKZxPy](https://klaraoppenheimer-my.sharepoint.com/:f/g/personal/zobel_klara-oppenheimer-schule_de/EloJcg3VjD5Jj6SH8wqRpCoBoDRPBAFW_CZldWGxUlDkRw?e=rKZxPy)



## Übungsaufgabe zum Strategy Pattern

### Transferaufgabe Rollenspiel

#### Aufgaben:

Below you'll find a mess of classes and interfaces for an action adventure game. You'll find classes for game characters along with classes for weapon behaviors the characters can use in the game. Each character can make use of one weapon at a time, but can change weapons at any time during the game. Your job is to sort it all out...



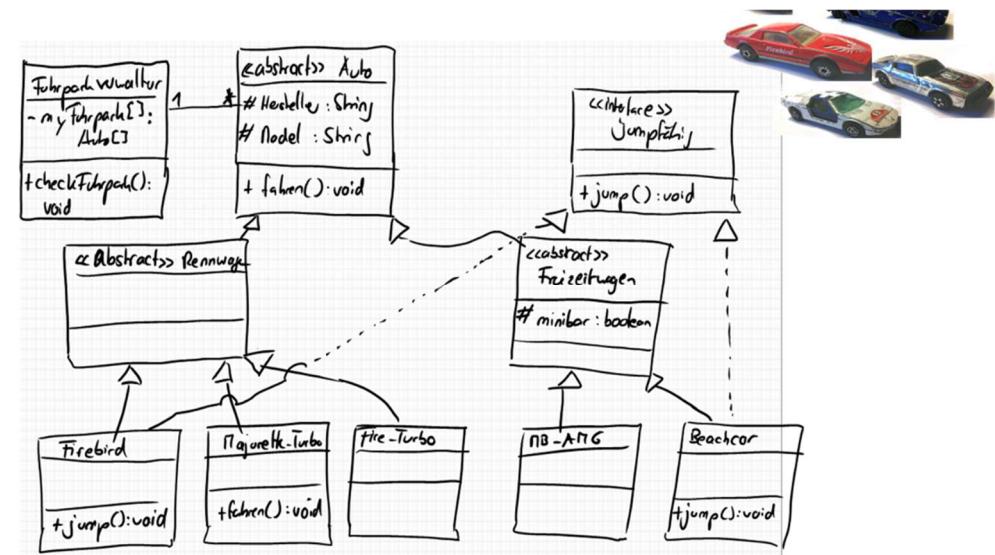
```

setWeapon(WeaponBehavior w) {
    this.weapon = w;
}
  
```

### Transferaufgabe Fuhrpark

#### Aufgaben:

- Passen Sie das obige Beispiel entsprechend des Strategy Pattern an. Erstellen Sie hierzu ein neues Klassendiagramm.
- Setzen Sie anschließend das vereinbarte Klassendiagramm in PHP um.
- Weisen Sie sodann den Autos zur Laufzeit unterschiedliches Fahrverhalten zu.

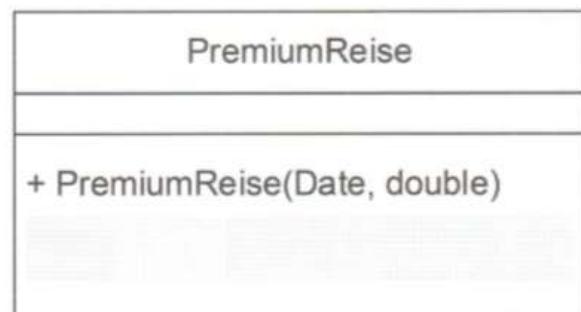
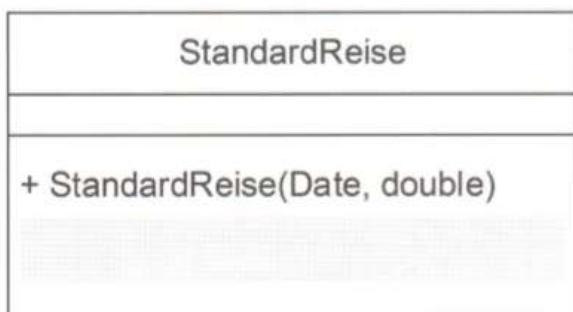
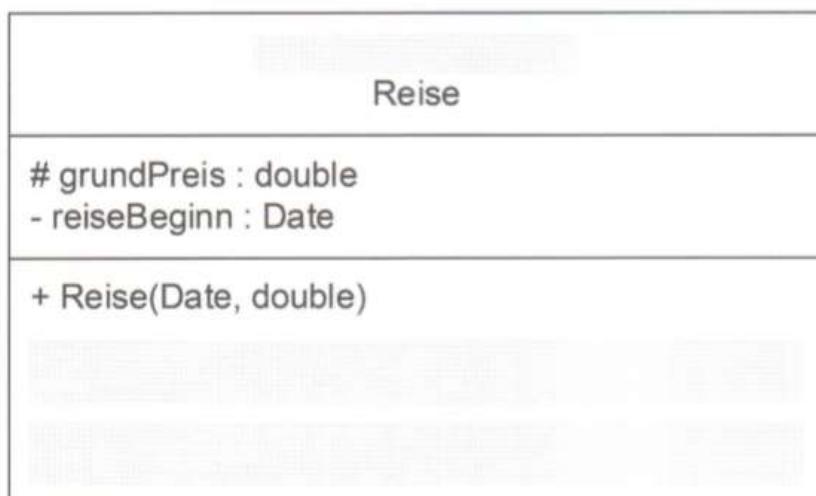


IHK-Aufgabe: Strategy Pattern

**2. Handlungsschritt (25 Punkte)**

Die Nord-West Reederei AG will ihren Kunden verschiedene Reisekategorien anbieten. Begonnen wird mit den Reisekategorie-Klassen *StandardReise* und *PremiumReise*. Zur Verwaltung soll ein objektorientiertes Programm entwickelt werden.

- a) Ein grober, noch unvollständiger Entwurf liegt in Form eines UML-Klassendiagramms bereits vor. Die Instanzvariablen werden über den Konstruktor initialisiert. Alle Reiseklassen sollen folgende öffentliche Methoden beinhalten:  
*tageBestimmen*: ermittelt die Anzahl der Tage bis zum Reisebeginn und gibt sie als ganze Zahl zurück.  
*preisBerechnen*: kalkuliert mit unterschiedlichen Algorithmen für Standard- und Premiumreisen den Reisepreis aus dem Grundpreis und gibt ihn als Dezimalzahl zurück.
- aa) Ergänzen Sie die Beziehungen zwischen den Klassen und kennzeichnen Sie die Klasse *Reise* als nicht instanzierbar. 3 Punkte
- ab) Ergänzen Sie die Methoden *tageBestimmen* und *preisBerechnen*. 4 Punkte



- b) Die Gebühren für die Stornierung von Standardreisen und Premiumreisen werden nach unterschiedlichen Algorithmen berechnet:

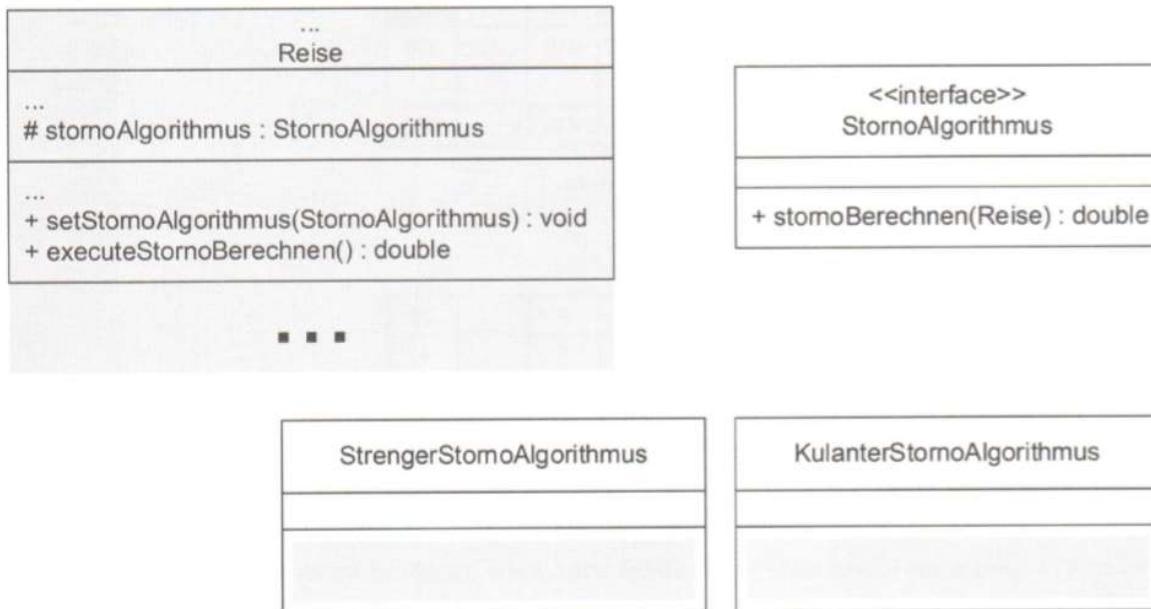
- Der strenge Algorithmus wird bei der Stornierung von Standardreisen angewendet.
- Der kulante Algorithmus wird sowohl bei der Stornierung von Premiumreisen als auch bei der Stornierung von Standardreisen, die von Stammkunden gebucht wurden, angewendet.

Die zu implementierende öffentliche Methode *stornoBerechnen* soll flexibel eingebunden werden.

Das folgende UML-Klassendiagramm wurde nach dem Strategie-Entwurfsmuster vorbereitet.

Vervollständigen Sie das UML-Klassendiagramm, indem Sie Folgendes ergänzen:

- Die Beziehungen zwischen den Klassen
- Die Methoden in den Klassen *StrengerStornoAlgorithmus* und *KulanterStornoAlgorithmus*



Ergänzen Sie ...

die Beziehungen zwischen den Klassen

die Methoden in den Klassen *StrengerStornoAlgorithmus* und *KulanterStornoAlgorithmus*.

6 Punkte

- c) Stellen Sie folgende Methoden/Methodenaufrufe in Pseudocode dar.

Entwerfen Sie jeweils den Quelltext in einer Ihnen bekannten Programmiersprache für ...

- ca) die Methode *executeStornoBerechnen*.

4 Punkte

- 
- 
- 
- 
- cb) den Konstruktor der Klasse *StandardReise*, in dem auch der kulante Stornoalgorithmus gesetzt wird.
- cc) den Methodenaufruf, der dafür sorgt, dass beim StandardReise-Objekt *hurtigSR* der kulante Stornoalgorithmus zur Anwendung kommt.

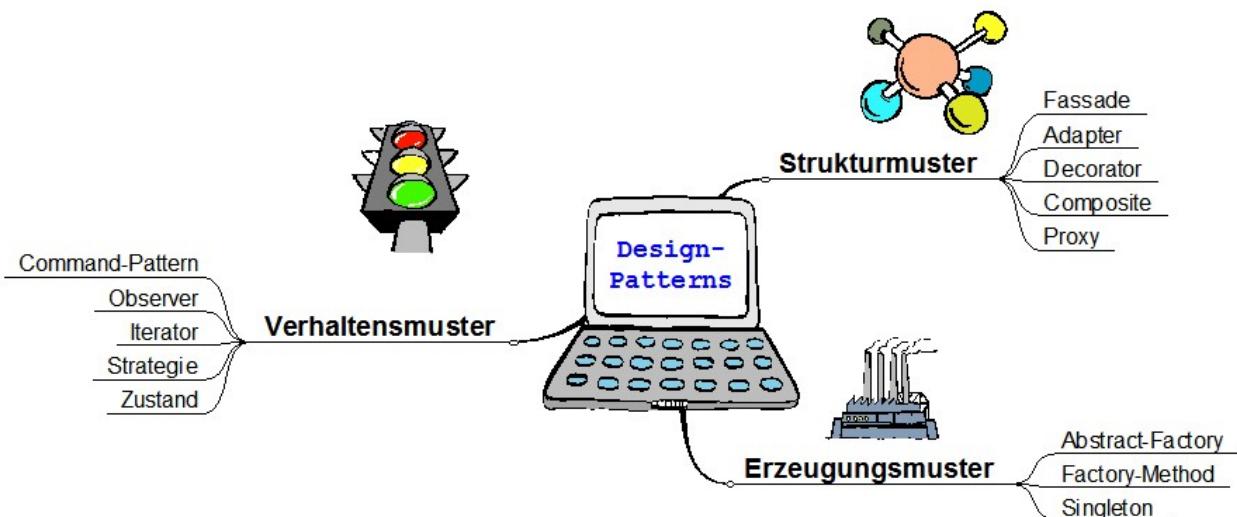
## 5.2 Weitere Design Pattern

### Wozu benötigen wir Entwurfsmuster?

Entwurfsmuster lösen bekannte, wiederkehrende Entwurfsprobleme.

Sie fassen Design- und Architekturwissen in komakter und wiederverwertbarer Form zusammen. Sowohl Software-Entwicklern als auch Software-Architekten bieten Entwurfsmuster wertvolle Unterstützung bei der Wiederverwendung erprobter Designentscheidungen. Sie geben Hinweise, wie Sie vorhandene Entwürfe flexibler, verständlicher oder auch performanter machen können. In komplexen Software-Projekten kann der angemessene Einsatz von Mustern das Risiko von Entwurfsfehlern deutlich senken.<sup>8</sup>

### Auswahl von Design-Pattern



### Arbeitsauftrag

1. Finden Sie sich in der Ihnen zugewiesenen Gruppe zusammen und überfliegen Sie die Informationen auf folgender Seite: <https://designpatternsphp.readthedocs.io/en/latest/> zu dem von Ihnen gewählten Pattern.
2. Besprechen Sie anschließend gemeinsam, wie Sie folgendes Ziel innerhalb der nächsten 3 Unterrichtsstunden erreichen wollen:

*Bereiten Sie eine Schulungssequenz vor, in der Sie innerhalb von ca. 30 Minuten Ihren Mitschülern anhand eines einfachen und leicht verständlichen Beispiels mit klaren Worten das Ihnen zugewiesene Design Pattern erklären. Erstellen Sie zudem eine passende Übung, die Sie in dieser Zeit durchführen und betreuen. Fertigen Sie hierzu auch ein einseitiges Handout an.*

<sup>8</sup> Vgl. Eilebrecht/Starke, Patterns kompakt.

## 6. Allerlei – was noch fehlte

