

# Problemorientiertes Lernen als didaktische Methodik zur Vermittlung fortgeschrittener Konzepte der Softwareentwicklung

Von Frederic Birwe  
Matrikel-Nr.: 1271201

# 1 Inhaltsverzeichnis

1	Inhaltsverzeichnis .....	2
2	Einleitung.....	4
3	Theoretischer Hintergrund .....	4
3.1	Kompetenzorientierung.....	4
3.2	Lernzieltaxonomie .....	5
3.3	problemorientiertes Lernen .....	6
4	Praktischer Versuch .....	6
4.1	Ziele der Auswertung.....	6
4.1.1	faktische Vermittlung von Konzepten der Softwareentwicklung .....	6
4.1.2	Lernerfahrung und Motivierung zur weiteren Beschäftigung mit den vermittelten Themen.....	7
4.2	Beschreibung des Versuchsaufbaus .....	7
4.3	Versuchsteilnehmer.....	7
4.4	Gewählte Programmiersprache.....	7
4.5	Rahmenbedingungen im Versuch unter Verwendung des problemorientierten Lernens .....	8
4.6	Rahmenbedingungen im Versuch unter Verwendung klassischer Didaktik .....	8
4.7	Thema: Objektorientierung .....	9
4.8	Kontrollthema: functional programming.....	11
4.9	Prüfung des Erlernten .....	13
4.9.1	Kontrolltest .....	13
4.9.2	Teilnehmende Beobachtung.....	13
4.9.3	Feedbackgespräch .....	13
5	Ergebnisse.....	14
5.1	Beobachtungen und Feedback .....	14
5.1.1	Selbsteinschätzung des Lernerfolgs.....	14
5.1.2	Erzeugung eines positiven Lernumfelds und Langzeitmotivierung .....	14
5.1.3	Persönliche Erfahrung mit dem didaktischen Ansatz .....	14
5.2	Auswertung der Kontrolltests.....	15
5.2.1	Auswertung in Hinblick auf vermitteltes Thema .....	15
5.2.2	Auswertung in Hinblick auf den verwendeten didaktischen Ansatz.....	16
5.2.3	Auswertung nach Taxonomieniveau .....	17
5.3	Gesamtergebnis.....	21
5.4	Kritische Reflektion.....	21
6	Literaturverzeichnis .....	23

7	Abbildungsverzeichnis .....	24
8	Tabellenverzeichnis .....	25
9	Anhang.....	26

## 2 Einleitung

Erfolgreiche Softwareentwicklung setzt eine Mischung aus faktischem Wissen, prozessualen Fähigkeiten und Fähigkeiten zur kreativen Nutzung und Adaption bekannter Konzepte voraus. Gleichzeitig umfasst der Bereich der Softwareentwicklung eine Vielzahl von unterschiedlichsten Themengebieten und Teilaspekten, dass eine umfängliche Ausbildung angehender Softwareentwickler und Softwareentwicklerinnen nicht möglich erscheint. Das didaktische Konzept des problemorientierten Lernens erlaubt es den Schülern und Schülerinnen selbst die zu erlernenden Inhalte auszuwählen. Gleichzeitig orientiert sich das problemorientierte Lernen immer an einem praktisch zu lösenden Problem, wodurch alle der oben beschriebenen Wissensdimensionen angesprochen werden. Im Rahmen dieses Experimentes soll geklärt werden, ob das problemorientierte Lernen sinnvoll in der Didaktik der Softwareentwicklung eingesetzt werden kann.

## 3 Theoretischer Hintergrund

Nachfolgend werden einige theoretische Hintergründe erläutert, auf denen das Konzept des Versuches sowie die konkrete Ausgestaltung fußt. Dabei werden zunächst die Begriffe der Kompetenzorientierung und der Lernzieltaxonomie erläutert. Diese theoretischen Hintergründe bilden die Basis dafür, die Kompetenzen, welche die Schüler nach einer Lerneinheit erworben haben sollen, zu klassifizieren, zu messen und schließlich auch zu bewerten. Abschließend wird das Konzept des problemorientierten Lernens erläutert, welches als Grundmodell für den hier durchgeführten Versuch dient.

### 3.1 Kompetenzorientierung

Kompetenzorientiertes Lehren und Lernen versucht dem zu Unterweisenden Kompetenzen zu vermitteln, mit denen er selbstständig in der Lage ist, Fragestellungen aus einem bestimmten Fachbereich zu lösen.

Kompetenz wird durch Weinert [1](13 Abs. 2.1, S346) definiert als „die bei Individuen verfügbaren oder durch sie erlernten kognitiven Fähigkeiten und Fertigkeiten, um bestimmte Probleme zu lösen, sowie die damit verbundenen motivationalen, volitionalen und sozialen Bereitschaften und Fähigkeiten, um die Problemlösungen in variablen Situationen erfolgreich und verantwortungsvoll nutzen zu können.“. Der Begriff der Kompetenz geht also über die reine Vermittlung von Wissen oder Fertigkeiten hinaus, sondern will diese Bausteine kombiniert vermitteln mit dem Ziel, dass ein Individuum danach selbstständig in der Lage ist, seine Kompetenz zur angemessenen Problemlösung anzuwenden.

Kompetenzziele lassen sich transparent und nachvollziehbar in sogenannten Transparenzrastern festhalten und ihre Erreichung kann anhand dieser definierten Ziele gemessen werden (siehe dazu Abs. 3.2).


Der Begriff der Kompetenzorientierung lässt sich gut auf die Didaktik der Softwareentwicklung anwenden. Eine hohe Kompetenz im Feld der Softwareentwicklung lässt sich nur erreichen, wenn sowohl die theoretischen Hintergründe (Eigenschaften von Programmiersprachen, Datentypen, Algorithmen, Entwurfsmuster, usw.) als auch prozessuale Fähigkeiten beherrscht werden. Schließlich ist in der Softwareentwicklung eine Fähigkeit zur Abstraktion von Problemen und Anwendung von kognitiven und prozessualen Fähigkeiten zur Problemlösung notwendig.

### 3.2 Lernzieltaxonomie

Entscheidend zur Festlegung der Ziele dieses Versuchs ist die möglichst eindeutige Definition von Lernzielen. Diese Lernziele dienen nachfolgend sowohl zur Ausarbeitung von Lehreinheiten als auch zur Messung der vermittelten Kompetenzen.

In der Didaktik werden zur Objektivierung und Planung von Lernzielen sogenannte Lernzieltaxonomien genutzt. Diese bieten einen Rahmen zur Einordnung der geplanten Lernziele.

Zur Konzeptionierung dieses Versuchs wurde auf die Lernzieltaxonomie nach Anderson und Krathwohl zurückgegriffen [2].



Stufe	Lernziele formulieren
6 Kreieren	„plant“, „produziert“, „generiert“, ...
5 Evaluieren	„überprüft“, „beurteilt“, „entscheidet“, ...
4 Analysieren	„differenziert“, „unterscheidet“, „findet Analogien“, ...
3 Anwenden	„nutzt das Modell XY/das Vorgehen PQ, um ein Problem zu lösen“, ...
2 Verstehen	„erläutert“, „erklärt“, „findet Beispiele“, „subsumiert“, „generalisiert“, ...
1 Erinnern	„kennt“, „nennt“, „zählt auf“, ...

Abbildung 1 Darstellung der Lernzieltaxonomie nach Anderson und Krathwohl, Bild übernommen aus [2]

Anderson und Krathwohl teilen Lernziele hier in die folgenden sechs Kategorien auf:

1. Erinnern
2. Verstehen
3. Anwenden
4. Analysieren
5. Beurteilen
6. Erschaffen

Die Stufen sind hierarchisch sortiert, das bedeutet, das Erreichen eines Lernziels einer höheren Stufe ist als besser zu bewerten als das Erreichen eines Lernziels einer niedrigeren Stufe.

Ab der dritten Stufe (anwenden) aufwärts gehen Anderson und Krathwohl davon aus, dass der Schüler hier eine Kompetenz erworben hat, das erlernte Wissen also in einem gegebenen Rahmen sinnvoll selbständig anwenden kann (siehe dazu Abschnitt 3.1).

Einher mit den sechs formulierten Stufen der Lernziele gehen Operatoren, welche sich besonders für die Formulierung von Lernzielen auf dem entsprechenden Niveau eignen. In der folgenden Tabelle findet sich eine Auswahl von Operatoren zugeordnet zu den einzelnen Niveaustufen nach Anderson und Krathwohl. [3] [4]

Kompetenzstufe	Operatoren
erinnern	Aufzählen, benennen, wiederholen, beschreiben
Verstehen	Erklären, veranschaulichen, erläutern
Anwenden	Definieren, anwenden, vervollständigen
Analysieren	Untersuchen, vergleichen
Beurteilen	Begründen, bewerten, auswerten
Erschaffen	Entwerfen, planen, gestalten

*Tabelle 1 Kompetenzstufen nach Anderson und Krathwohl mit entsprechenden Operatoren*

### 3.3 problemorientiertes Lernen

Der für diesen Versuch gewählte Ansatz orientiert sich am Konzept des problemorientierten Lernen nach Don Woods. Beim problemorientierten erfolgt die Lehre in der Regel in Form von Lerngruppen. Der Lerngruppe wird dabei eine Problemaufgabe gestellt, welche von der Lerngruppe selbstständig in der vorgegebenen Zeit gelöst werden soll. Die zur Lösung der Problemaufgabe notwendigen Kompetenzen sind in der Lerngruppe zu Beginn der Bearbeitung nicht vorhanden. Im Gegenteil ist es elementarer Bestandteil des problemorientierten Lernens, dass die Lerngruppe selbst festlegt, welche Inhalte in welcher Reihenfolge und in welchem Umfang erlernt werden müssen, um die gestellte Aufgabe zu lösen. Die Lehrperson tritt in diesem Lernzusammenhang primär als Supervisor auf. Sachliche Inhalte werden von der Lehrperson nur in Ausnahmefällen und in reduziertem Umfang vermittelt [5].

Beim problemorientierten Lernen wählen die Schüler anhand selbst aus, wann sie welche Lerninhalte erlernen wollen. Um trotzdem sicherzustellen, dass die Schüler das von der Lehrperson geplante Lernziel erreichen, kommt der Ausgestaltung der Problemstellung besondere Wichtigkeit zu.

## 4 Praktischer Versuch

### 4.1 Ziele der Auswertung

Die Untersuchung soll in zwei Dimensionen ausgewertet werden. Die erste Dimension ist der Erfolg der Vermittlung von fachlichen Kenntnissen und Fähigkeiten in den gewählten Themen an die Probanden. Daneben soll weiterhin die Lernerfahrung sowie die Motivation zur weiteren selbstständigen Auseinandersetzung mit den vermittelten Themen erfragt werden.

#### 4.1.1 faktische Vermittlung von Konzepten der Softwareentwicklung

Die Art und Schwierigkeit der Aufgaben orientiert sich dabei an der Lernzieltaxonomie nach Anderson und Krathwohl (siehe: Abs. 3.2)

Die Fragen des Kontrolltests umfassen Fragen, die auf die unterschiedlichen Lernzieldimensionen abzielen und so testen, wie erfolgreich der Weg der Vermittlung war und ob einzelne Wissens- oder kognitive Prozessdimensionen besonders positiv oder negativ durch einen bestimmten Weg der Vermittlung beeinflusst werden). Eine detaillierte Beschreibung der Messung findet sich in Abs. 4.9.1.

#### 4.1.2 Lernerfahrung und Motivierung zur weiteren Beschäftigung mit den vermittelten Themen

Wie in Abschnitt 3.1 erläutert, erfordert die Softwareentwicklung insbesondere Fähigkeiten der kreativen Anwendung bekannter Konzepte auf neue Szenarien. Diese höheren Level des Verständnisses (siehe: Abs. 3.2) sind ohne intrinsische Motivation des Schülers und ohne Vertiefung über die Unterrichtseinheiten hinaus nicht zu erreichen [1](12 Abs.2, S.314). Daher soll in einem zweiten Messverfahren betrachtet werden, inwieweit die Wahl der didaktischen Methode sich auf die intrinsische Motivation des Schülers auswirkt, sich weitergehend mit den Themen der Unterrichtseinheit auseinanderzusetzen. Eine detaillierte Beschreibung der Erfassung dieser Ergebnisse findet sich in Abschnitt 4.9.3.

#### 4.2 Beschreibung des Versuchsaufbaus

Der Versuch findet wie folgt statt: Pro Testperson werden zwei Termine im Abstand von ca. einer Woche durchgeführt. Jeder dieser Termine ist für eine Stunde ausgelegt. Innerhalb dieser Zeit wird jeweils eines der genannten Themen mit einem der beschriebenen didaktischen Prinzipien vermittelt.

Der Ablauf der Ergebnismessung wird in Abschnitt 4.9 beschrieben.

Um die Ergebnisse des Versuchs auf eine validere Basis zu stellen, werden die Tests pro Testperson jeweils über Kreuz durchgeführt. Das bedeutet, wenn Testperson A das Thema Objektorientierung anhand der Methode des problemorientierten Lernens vermittelt wurde, wird dieser Person das Thema funktionale Programmierung entsprechend mit einer klassischen Methode vermittelt. Bei Testperson B findet die Vermittlung genau umgekehrt statt. Ihr wird das Thema Objektorientierung mit einer klassischen Methode vermittelt und das Thema funktionale Programmierung mit Hilfe des problemorientierten Lernens.

#### 4.3 Versuchsteilnehmer

Zur Erprobung eines problemorientierten Ansatzes konnten zwei Testpersonen gewonnen werden.

Die Personen sind 51 und 43 Jahre alt. Beide sind männlich. Sie besitzen grundlegende Kenntnisse im Bereich Softwareentwicklung haben jedoch keine Expertise in den verwendeten Beispielthemen Objektorientierung und functional programming.

Das bedeutet konkret, sie beherrschen die folgenden Konzepte:

- Variablen
- Schleifen
- Bedingungen
- Funktionen
- Datentypen
- Arrays

Besitzen darüber hinaus aber wenig bis keine Kenntnisse im Bereich Softwareentwicklung. Beide arbeiten in technischen Bereichen, die hohe IT-Kompetenz aber nicht zwingend Kompetenz im Bereich Softwareentwicklung voraussetzen.

#### 4.4 Gewählte Programmiersprache

Als Programmiersprache der Übungen wird JavaScript gewählt. JavaScript bietet alle notwendigen Konstrukte, um das in diesem Versuch benötigte Niveau objektorientierte Programmierung sowie funktionaler Programmierung zu betreiben. Nichtsdestotrotz ist JavaScript keine objektorientierte Programmiersprache.

Hier wurde JavaScript gewählt, da bei den Teilnehmern bereits Vorkenntnisse vorhanden sind. Explizit objektorientierte Programmiersprachen wie Java oder C++ sind hingegen unter den Testpersonen nicht verbreitet. Sie wurden hier nicht gewählt, um nicht zusätzlich zu objektorientierten Konzepten auch eine neue Programmiersprache vermitteln zu müssen. Zudem wird als Beispiel zur Vermittlung der Konzepte ein Programm verwendet, das direkt im Browser ausgeführt werden kann und hier auch mit einer Visualisierung unterstützt wird (siehe Anhang 10 und Anhang 20). Dies lässt sich am einfachsten mit der bereits im Browser integrierten Programmiersprache JavaScript umsetzen.

#### 4.5 Rahmenbedingungen im Versuch unter Verwendung des problemorientierten Lernens

Zentrales Konzept des problemorientierten Lernens ist, dass die Schüler selbstständig entscheiden, welche Informationen zu welchem Zeitpunkt im Lernprozess benötigt werden. Daher werden für diesen Versuch Ihnen alle relevanten Informationen zu Beginn der Unterweisung übergeben.

Im Rahmen dieses Versuches ist dies jeweils ein beispielhaftes Programm, das unter Verwendung des jeweiligen Konzeptes erstellt wurde. Dieses Programm ist in JavaScript umgesetzt. Neben den JavaScript-Dateien, die für die Schüler zum Verständnis notwendig sind und in welchen diese arbeiten sollen, enthalten die Beispielprogramme noch weitere JavaScript-Dateien sowie HTML- und CSS-Dateien, welche die vom Schüler vorgenommenen Veränderungen visuell im Browser darstellen (Anhang 10 und Anhang 20).

Zusätzlich werden dem Schüler weitere Materialien sowie externe Informationsquellen zur Verfügung gestellt. Die genauen weiteren Materialien werden in den folgenden Abschnitten jeweils themenspezifisch genannt. Des Weiteren steht es den Schülern frei, sich weitere Informationen über das Internet einzuholen oder den Unterweiser um Erklärung von Konzepten zu bitten.

Schließlich werden die Schüler mit einer Aufgabenliste ausgestattet (Anhang 12 und Anhang 15). Diese Liste soll als Orientierung und Leitfaden dienen, über den die Schüler alle wichtigen Konzepte kennenlernen.

Die Schüler werden zu Beginn des Versuchs detailliert über die Rahmenbedingungen aufgeklärt.

Die Schüler sollen die Lösungen der Aufgaben eigenständig erarbeiten. Sie legen die Reihenfolge der Bearbeitung fest. Zudem lassen die meisten Aufgaben mehrere Lösungswege zu. Auch hier entscheiden die Schüler über die Wahl des Lösungsweges. Der Unterweiser greift nur auf Nachfrage ein und erläutert einzelne Zusammenhänge und Teilkonzepte, gibt aber keine konkrete Hilfestellung zur Lösung der Aufgaben.

#### 4.6 Rahmenbedingungen im Versuch unter Verwendung klassischer Didaktik

Im Versuch unter Verwendung klassischer Didaktik wird das Thema in Form einer Präsentation des Unterweisenden vermittelt. Dafür wird eine Bildschirmpräsentation sowie ein beispielhaftes Programm verwendet (Anhang 8, Anhang 14, Anhang 17). Um beide Versuche möglichst vergleichbar zu halten, ist das beispielhafte verwendete Programm identisch mit dem Programm, das zur problemorientierten Vermittlung genutzt wird.

Die Unterweisung erfolgt weitestgehend als Vortrag des Unterweisenden. Im Verlauf der Unterweisung werden an verschiedenen Stellen Fragen an die Schüler gestellt, die das Verständnis der Konzepte abfragen und so die eine selbstständige Auseinandersetzung mit den Themen notwendig machen. Der Schüler wird zudem ermutigt, sich aktiv an der Unterweisung zu beteiligen, indem er Fragen stellt oder in die Diskussion mit dem



Unterweisenden kommt. Dies kann vom Unterweisenden aus Zeitgründen unter Umständen eingeschränkt werden.

Nach der theoretischen Vorstellung der Konzepte in Form einer Bildschirmpräsentation mit Beispielen wird das Erlernete in einer praktischen Anwendung vertieft. Dazu steht das selbe Beispielprogramm zur Verfügung, dass auch im problemorientierten Ansatz verwendet wird. Unterweiser und Schüler bearbeiten nun im Dialog die Aufgabenstellungen aus dem problemorientierten Ansatz, solange es der gesetzte Zeitrahmen ermöglicht. Die Reihenfolge der Bearbeitung gibt der Unterweiser vor.

#### 4.7 Thema: Objektorientierung

Als exemplarisches fortgeschrittenes Konzept, dass vermittelt werden soll, wurde hier die objektorientierte Programmierung ausgewählt. Dabei sollen im Rahmen einer einstündigen Einheit die folgenden Themen vermittelt werden:

- Idee der Objektorientierung
- Klassen
- Eigenschaften und Methoden
- Konstruktoren
- Vererbung

Tiefergehende Themen wie Sichtbarkeit, Getter/Setter, Polymorphie oder Interfaces, usw. werden auf Grund des beschränkten Umfangs der Unterweisung nicht bearbeitet.

Konkret sollen im Idealfall die folgenden Lernziele erreicht werden:

Nr.	Lernziel	Thema	Klassifizierung nach Anderson und Krathwohl
1.1	Die Schüler sind in der Lage, die folgenden Begriffe zu benennen <ul style="list-style-type: none"><li>- Klasse</li><li>- Objekt</li><li>- Eigenschaft</li><li>- Methode</li><li>- Konstruktor</li></ul>	Grundlagen OOP	Erinnern
1.2	Die Schüler sind in der Lage, in einem gegebenen Programm die folgenden Bestandteile zu identifizieren <ul style="list-style-type: none"><li>- Klasse</li><li>- Objekt</li><li>- Eigenschaft</li><li>- Methoden</li><li>- Konstruktor</li></ul>	Grundlagen OOP	Verstehen
1.3	Die Schüler sind in der Lage, gegebene Beispiele korrekt den folgenden Konstrukten zuzuordnen <ul style="list-style-type: none"><li>- Klasse</li><li>- Objekt</li></ul>	Grundlagen OOP	Anwenden

	<ul style="list-style-type: none"> <li>- Eigenschaft</li> <li>- Methode</li> </ul>		
2.1	Die Schüler sind in der Lage, anhand eines Beispiels aus der realen Welt eine sinnvolle Objektstruktur zu entwickeln	Vererbung	Erschaffen
2.2	Die Schüler sind in einem gegebenen Programm in der Lage zu erkennen, ob eine Klasse eine Subklasse einer anderen Klasse ist oder nicht	Vererbung	Beurteilen
2.3	Die Schüler sind in der Lage, bei Klassen und Subklassen Eigenschaften und Methoden einer Klasse vollständig aufzulisten	Vererbung	Beurteilen
2.4	Die Schüler kennen das Schlüsselwort „instanceof“ und können seine Funktion beschreiben.	Vererbung	Verstehen

*Tabelle 2 Lernziele objektorientierte Programmierung*

Diese Lernziele beschreiben das theoretische Maximum, dass von den Testpersonen nach der Unterrichtseinheit beherrscht werden kann. Es wird nicht erwartet, dass die Testpersonen nach der Unterrichtseinheit in der Lage sind, alle diese Lernziele zu erfüllen. Vielmehr dient der Anteil der erreichten Lernziele als Kennwert dafür, wie erfolgreich die Konzepte vermittelt wurden.

Grundlegende Kompetenz im Thema objektorientierte Programmierung setzt Verständnis in den folgenden Bereich voraus:

- Neue Syntax
- Anwendung neuer Programmierkonzepte
- Modellierung von Realweltproblemen mit objektorientierten Konzepten

Als neue Syntax in der Programmiersprache sind hier vor allem die folgenden Schlüsselwörter zu nennen:

- class
- new
- constructor
- this
- instanceof
- extends
- super

Im Bereich der neuen Programmierkonzepte soll im Kern verstanden werden, welche Bestandteile (Parameter, Methoden, Konstruktoren) eine Klasse hat und wie man Klassen und Objekte in Programmen nutzt.

Die Modellierung von Problemen aus der realen Welt mit Konzepten der objektorientierten Programmierung schließlich erfordert die Fähigkeit der Abstraktion und ein tiefergehendes Verständnis der Idee hinter dem objektorientierten Ansatz.

Die Lösungsstruktur in objektorientierter Programmierung orientiert sich sehr stark an der realen Welt. Daher wird für diesen Versuch die Hypothese aufgestellt, dass sich die benötigte Abstraktionsfähigkeit hier vergleichsweise schnell erreichen lässt, da sich Lösungsstrukturen sehr einfach aus der realen Welt ableiten lassen.

#### 4.8 Kontrollthema: functional programming

Als zweites Thema zur Gegenprobe wurde das Thema functional programming ausgewählt.

Dabei wurden die folgenden Unterthemen vermittelt:

- Side Effects/Pure und Impure Functions
- Arrow Functions
- Higher Order Functions
- Function Composing
- Immutable Data

Diese Themen sind in den folgenden Lernzielen abgebildet:

Nr.	Lernziel	Thema	Klassifizierung nach Anderson und Krathwohl
1.1	Die Schüler können die folgenden Begriffe beschreiben: <ul style="list-style-type: none"> <li>- Side Effect</li> <li>- Pure Function</li> <li>- Impure Function</li> <li>- Arrow Function</li> <li>- Higher Order Function</li> <li>- Function Composing</li> <li>- Immutable Data</li> </ul>	Alle	Erinnern
2.1	Die Schüler sind in der Lage, die Eigenschaften zu benennen, durch die eine Funktion zu einer Pure Function wird.	Pure Function/Side Effects	Verstehen
2.2	Die Schüler sind in der Lage, bei einer gegebenen Funktion zu unterscheiden, ob diese Pure oder Impure ist.	Pure Function	Beurteilen
2.3	Die Schüler sind in der Lage, zu beschreiben, was ein Side Effect ist	Side Effects	Verstehen
2.4	Die Schüler können eine impure Function so umbauen, dass sie eine pure function wird	Pure Function/Side Effects	Erschaffen
3.1	Die Schüler können Parameter und Rückgabewert einer Arrow-Function korrekt identifizieren	Arrow Function	Untersuchen

3.2	Die Schüler können eine normale Funktion in eine Arrowfunktion umbauen und umgekehrt	Arrow Function	Erschaffen
4.1	Die Schüler können erläutern, welche Eigenschaften eine Higher Order Function definieren	Higher Order Functions	Verstehen
4.2	Die Schüler sind in der Lage, eine gegebene Higher Order Function (zum Beispiel map oder filter) anzuwenden, um eine Problemstellung zu lösen	Higher Order Function	Erschaffen
5.1	Die Schüler sind bei der Verwendung von Function Composing in Lage zu erkennen, welche Operationen in welcher Reihenfolge durchgeführt werden.	Function Composing	Analysieren
5.2	Die Schüler sind in der Lage, ein Function Composing so anzupassen, dass bestimmte Operationen nicht mehr oder anders ausgeführt werden.	Function Composing	Erschaffen
6.1	Die Schüler sind in der Lage zu beschreiben, welche Eigenschaften Immutable Data ausmachen	Immutable data	Verstehen
6.2	Die Schüler sind in der Lage zu erkennen, ob ein gegebener Codeabschnitt mutable data enthält oder nicht.	Immutable Data	Analysieren
6.3	Die Schüler sind in der Lage, bei einfachen Datentypen (keine Arrays, Objekte, etc.), einen gegebenen Code so anzupassen, dass er keine mutable data mehr enthält	Immutable data	Erschaffen

*Tabelle 3 Lernziele funktionale Programmierung*

Die Anwendung von Functional Programming setzt, anders als objektorientierte Programmierung, wenig bis keine neue Syntax voraus. In diesem Versuch werden als neue Syntax Arrow-Funktionen eingeführt. Diese sind in der funktionalen Programmierung weit verbreitet, aber zur Anwendung nicht zwingend notwendig.

Funktionale Programmierung gibt hingegen sehr stark vor, wie Lösungen strukturiert sein sollen und welche Lösungsansätze im Sinne der funktionalen Programmierung sinnvoll sind und welche nicht.

Weiterführende Konzepte der funktionalen Programmierung, speziell Higher Order Functions, Function Composing und immutable Data setzen einen sicheren Umgang mit Datentypen und Funktionen im Allgemeinen voraus. Zudem lassen sich diese

Lösungskonzepte schwieriger als bei einem objektorientierten Ansatz aus der Realität ableiten. Im Rahmen dieses Versuches wird die Hypothese aufgestellt, dass Konzepte der funktionalen Programmierung weniger intuitiv erfassbar sind (speziell für nicht fortgeschrittene Softwareentwickler) und daher mit einem geringeren Lernerfolg zu rechnen ist.

## 4.9 Prüfung des Erlernten

### 4.9.1 Kontrolltest

Als Erfolgsmessung wird pro Übungseinheit und Testperson ein Test durchgeführt. Der Test erfolgt im Abstand ein bis zwei Tagen. Der Abstand zum Versuch dient dazu, dass nicht die kurzzeitig erinnerten, sondern die langfristig erlernten Fähigkeiten erfragt werden. Der Test ist für ca. 30 Minuten konzipiert.

Der Test ist so konzipiert, dass er erwartungsgemäß nicht vollständig gelöst werden kann. Stattdessen soll anhand der unterschiedlichen Lösungsgrade der Lernerfolg beziehungsweise der Erfolg didaktischen Methode bewertet werden.

Die Übungen im Test sollen sowohl die unterschiedlichen Themenbereiche als auch die verschiedenen Kompetenzen und Verständnisgrade abgefragt werden. Die Auswahl der Kontrollfragen orientiert sich an der in Abschnitt 3.2 beschriebenen Lernzieltaxonomie. Die konkreten Fragebögen sowie die Einordnung der Fragen im Sinn der Lernzieltaxonomie und die Bewertungsmatrix finden sich in Anhang 3.

### 4.9.2 Teilnehmende Beobachtung

Der eigentliche Test wird während der Durchführung dokumentiert. Der Unterweiser notiert sich während des Ablaufs grobe Zeitmarken (beispielsweise, wann wird die Bearbeitung welches Teilthemas begonnen). Zusätzlich notiert der Unterweiser Beobachtungen über das Verhalten des zu Unterweisenden, die ihm als besonders auffällig oder relevant erscheinen. In solchen Notizen kann zum Beispiel festgehalten werden, zu welchem Zeitpunkt der zu Unterweisende besonders motiviert wirkt, wenn er besonders in den Dialog mit dem Unterweiser geht oder andersherum wenn sich der zu Unterweisende besonders schwer tut. Die Ergebnisse werden zusammen

### 4.9.3 Feedbackgespräch

Zusätzlich zum Kontrolltest wird unmittelbar im Anschluss an den Versuch ein Feedbackgespräch mit der Testperson durchgeführt. Im Rahmen dieses Feedbackgesprächs soll herausgearbeitet werden, wie die Testperson selbst ihren Lernerfolg einschätzt und ob das Modell sich positiv auf Langzeitmotivation auswirkt. Dabei werden dem Schüler eine Reihe von Fragen gestellt, die in Interviewform beantwortet werden.

Die folgenden Fragen werden gestellt:

1. Wie zufrieden bist du mit dem eigenen Lernfortschritt?
2. Hast du das Gefühl, das Thema umfänglich verstanden zu haben?
3. Findest du das vermittelte Thema interessant?
4. Kannst du dir vorstellen, dich selbstständig weiter mit dem gelernten auseinanderzusetzen?
5. Hat dir die Teilnahme Spaß gemacht?
6. Was fandest du gut?
7. Was fandest du verbesserungswürdig?

Die Fragen sind bewusst als offene Fragen angelegt, um hier eine ausführliche und individuelle Antwort zu erhalten. Bei den geschlossenen Fragen wird der Schüler ebenfalls zu einer ausführlichen Antwort motiviert, etwa indem er gebeten wird, die eigene Einschätzung auf einer Skala von eins bis zehn einzuordnen. Die Fragen sind für alle Versuche, unabhängig von Thema und didaktischem Ansatz, identisch.

Die Fragen sollen Informationen zu drei Teilaspekten liefern:

1. Selbsteinschätzung des Lernerfolgs (Fragen 1 und 2)
2. Erzeugung eines positiven Lernumfelds und Langzeitmotivierung (Fragen 3,4 und 5)
3. Persönliche Erfahrung mit dem didaktischen Ansatz (Fragen 5,6 und 7)

Teilweise werden die Ergebnisse im Nachhinein mit dem faktischen Lernerfolg abgeglichen. So ist es beispielsweise denkbar, dass bei einem didaktischen Ansatz der Lernerfolg vom Schüler deutlich höher als eingeschätzt wird als er faktisch messbar ist.

## 5 Ergebnisse

### 5.1 Beobachtungen und Feedback

Nachfolgend werden die Ergebnisse des Feedbackgesprächs nach jeder Lehreinheit dargestellt und eingeordnet. Die Einordnung erfolgt anhand der in Abschnitt 4.9.3 beschriebenen Teilaspekte. Abschließend erfolgt eine Gesamtbewertung der Lerneinheit auch unter Zuhilfenahme der teilnehmenden Beobachtung.

#### 5.1.1 Selbsteinschätzung des Lernerfolgs

Beide Teilnehmer schätzen ihren Lernerfolg bei der Lerneinheit, die mit einem klassischen Ansatz durchgeführt wurde, als besser ein. Ein Teilnehmer sagt sogar aus, dass er das Thema der klassischen Unterweisung besser verstanden hat, obwohl er es als das anspruchsvollere Thema empfunden hat. Beide Teilnehmer sagen jedoch auch aus, dass sie bei beiden Themen zufrieden mit ihrem Lernfortschritt sind. In Abschnitt 5.2.1 wird darauf anhand der Kontrollfragebögen tiefer eingegangen. Ein Teilnehmer sagt aus, dass er nach seinem Empfinden beim klassischen Ansatz alle Themen in Teilen verstanden hat. Beim problemorientierten Ansatz hingegen hat er, nach seiner Aussage, nicht alle Themen verstanden, diejenigen, die er verstanden hat, aber umfänglich. Auf diese Aussage wird in 5.2.2 im Kontext der Kontrollfragebögen genauer eingegangen.

#### 5.1.2 Erzeugung eines positiven Lernumfelds und Langzeitmotivierung

Beide Teilnehmer empfanden die jeweiligen Themen, unabhängig vom gewählten didaktischen Ansatz, als interessant. Als interessanteres Thema wurde jeweils von beiden Teilnehmern, unabhängig vom gewählten didaktischen Ansatz, die objektorientierte Programmierung genannt. Beide Teilnehmer sagten aus, dass sie sich grundsätzlich vorstellen können, sich auch selbstständig weiter mit den Themen auseinanderzusetzen. Dies wäre aber jeweils abhängig davon, ob es aktuellen Bedarf zur Verwendung gäbe. Eine rein intrinsisch motivierte Auseinandersetzung mit den Themen ist nicht zu erwarten.

#### 5.1.3 Persönliche Erfahrung mit dem didaktischen Ansatz

Beide Teilnehmer nannten den klassischen Ansatz als den für sie angenehmeren Ansatz. Dabei wurde mehrfach darauf hingewiesen, dass die Erarbeitung von Wissen im problemorientierten Ansatz als Prüfungssituation und daher als stressend empfunden

wurde. Dies kann sowohl am gewählten Ansatz als auch an der Ausgestaltung der Situation durch die Lehrperson liegen. Der klassische Ansatz wurde hingegen als entspannter und angenehmer empfunden. Die Teilnehmer sagten dazu aus, dass die Lernsituation auch deshalb angenehmer war, weil sie weniger eigene Handlung erfordert und mehr „zurücklehnen“ ermöglicht. Wenn es gelang, bei den Unterweisungen in klassischer Didaktik in den Dialog mit den Teilnehmern zu kommen, wurde dies als angenehm und kooperativ empfunden. In vergleichbaren Situationen in den Unterweisungen mit problemorientiertem Ansatz wurden diese ebenfalls als die stärkeren Momente der Unterweisung empfunden, blieben jedoch in der Qualität der Atmosphäre hinter jenen bei klassischer Didaktik zurück. Ebenfalls herausgehoben starke Momente in der problemorientierten Herangehensweise konnten erreicht werden, wenn die Teilnehmer selbst kleine Änderungen am Code vornehmen und das Ergebnis direkt im Browser wahrnehmen konnten. Diese Ergebnisse ließen sich allerdings nur bei kleinen Anpassungen am Code beobachten. Komplexere Aufgabenstellungen mit mehreren Schritten bieten dieses direkte Feedback so nicht. Schließlich sorgte es teilweise für Irritationen, dass in den verwendeten Beispielen nach wie vor programmierte Teile vorhanden sind, die über das Niveau der Teilnehmer hinausgehen und nicht angepasst werden sollen. Die angesprochenen Teile des Codes sorgen dafür, dass die Programmierung der Teilnehmer im Browser grafisch dargestellt wird.

## 5.2 Auswertung der Kontrolltests

Nachfolgend werden die Ergebnisse der Kontrolltests dargelegt und ausgewertet

Die dargelegten Ergebnisse ergeben sich aus Anhang 2, Anhang 3, Anhang 4, Anhang 5 und Anhang 7

### 5.2.1 Auswertung in Hinblick auf vermitteltes Thema

Zunächst soll ausgewertet werden, wie viele Punkte pro Thema erreicht wurden. Weichen die erreichten Punktezahlen zu stark voneinander ab, kann dies ein Hinweis darauf sein, dass die gewählten Themen einen zu unterschiedlichen Schwierigkeitsgrad haben, Fragebogen oder Unterweisung von zu unterschiedlicher Qualität oder zu unterschiedlichem Schwierigkeitsgrad sind, und so weiter.

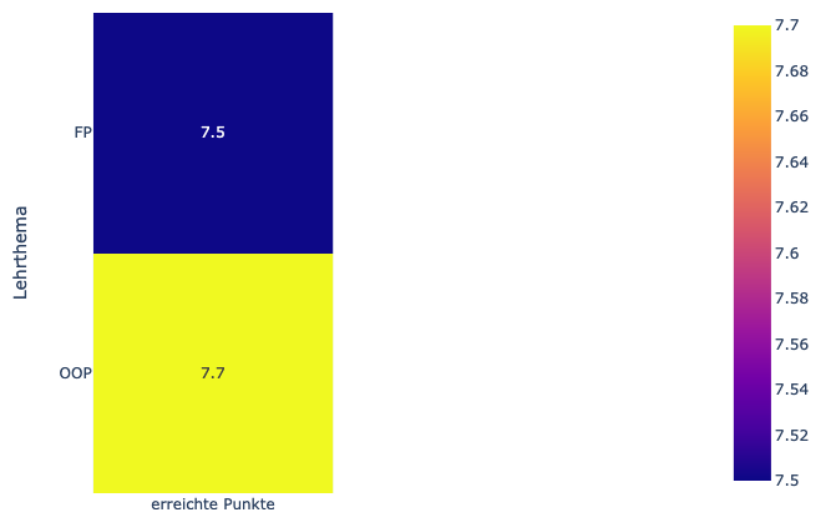


Abbildung 2 Erreichte Punkte, aufsummiert nach Thema

Abbildung 2 zeigt, dass für das Thema funktionales Programmieren insgesamt 7,5 Punkte erreicht wurden, für das Thema objektorientierte Programmierung insgesamt 7,7 Punkte. Die aufsummierten Ergebnisse liegen also sehr nah beieinander, sodass davon ausgegangen werden kann, dass die genannten Qualitätskriterien erreicht werden konnten.

Da pro Thema zwei Fragebögen mit je neun Fragen ausgefüllt werden sollten, können also pro Thema maximal 18 Punkte erreicht werden. Somit ergeben sich die folgenden Erreichungsgrade pro Thema

- Funktionales Programmieren: 0,4167
- Objektorientiertes Programmieren: 0,43

Dies kann also Hinweis darauf interpretiert werden, dass der gewählte Lehrinhalt deutlich zu umfangreich für die zur Verfügung stehende Zeit ist und in zukünftigen Versuchen reduziert werden sollte.

### 5.2.2 Auswertung in Hinblick auf den verwendeten didaktischen Ansatz

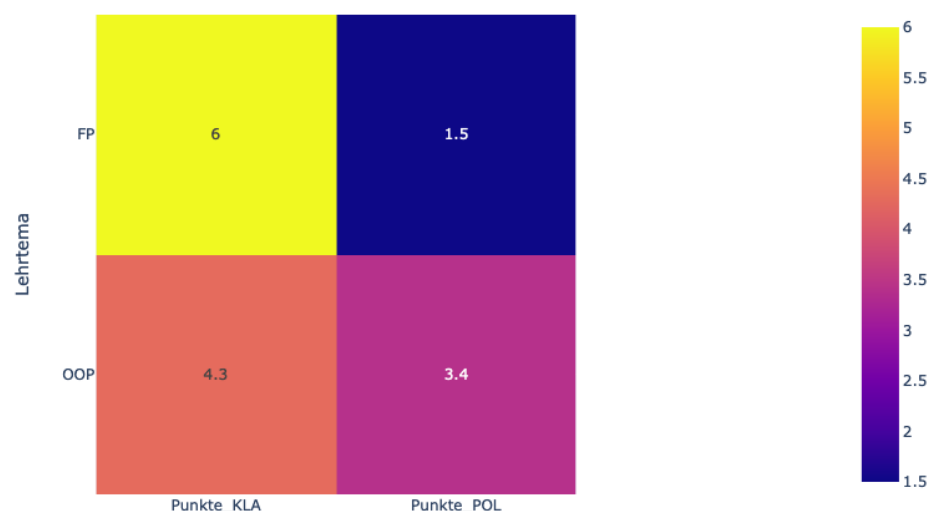


Abbildung 3 erreichte Punkte in den Kontrollfragebogen, aufsummiert nach Lernthema und didaktische Ansatz (eigene Darstellung)

Abbildung 3 zeigt die in den Kontrollfragebogen erreichten Punkte, jeweils aufsummiert nach Lernthema und didaktischem Ansatz. Die Abbildung lässt die folgenden Beobachtungen zu:

Zunächst kann festgehalten werden, dass bei jedem Lernthema jeweils mehr Punkte bei der Vermittlung mit Hilfe von klassischen didaktischen Mitteln erzielt wurde. Dies legt den Schluss nahe, dass im Rahmen dieses Versuches der problemorientierte Ansatz zu einer schlechter Kompetenzvermittlung geführt hat als die Vermittlung mit Hilfe klassischer Didaktik. Im Feedbackgespräch wurde die Aussage getroffen, dass der problemorientierte Ansatz zwar zu weniger Kenntnissen in der Breite, dafür aber zu tieferen Kenntnissen in den erlernten Themen geführt hat (siehe Abs. 5.1.1).



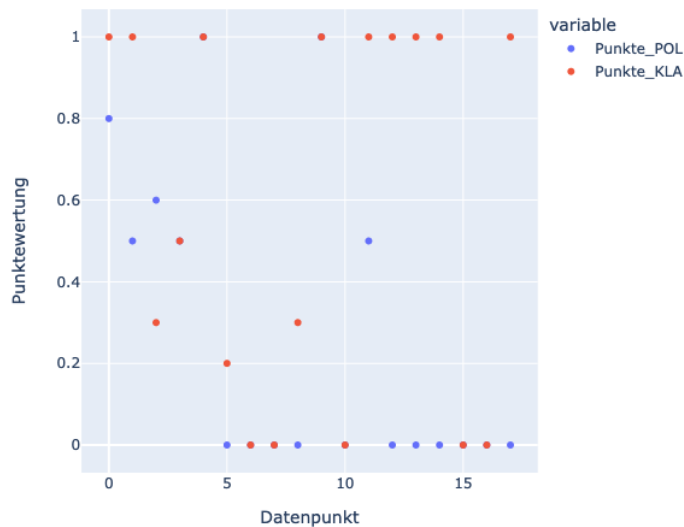


Abbildung 4 Punkteverteilung der Kontrollfragen nach didaktischem Ansatz (eigene Darstellung)

Mit Blick auf Abbildung 4 lässt sich diese Aussage nicht halten. Betrachtet man die Fragen, bei denen die Person, welche dieses Thema problemorientiert vermittelt bekam, mehr Punkte als die andere Person erzielen konnte, ist dies nur bei einer einzigen Frage der Fall. Bei neun Fragen konnten im klassischen Ansatz mehr Punkte erzielt werden. Bei acht Fragen wurden identische Punkte erreicht.

Als zweite Beobachtung aus Abbildung 3 lässt sich festhalten, dass die erzielten Punktwertungen zum Thema objektorientierte Programmierung recht nah beieinander liegen, während die beiden Ergebnisse zum Thema functional programming jeweils das beste und schlechteste der vier Ergebnisse sind. Dies könnte die These stützen, dass objektorientierte Programmierung das intuitiv besser zu erfassende Thema ist. Objektorientierte Konzepte lassen sich gut aus der realen Welt ableiten und visuell erfassen lässt, während sich funktionale Konzepte ohne nähere Erklärung schlechter erfassen lassen.

### 5.2.3 Auswertung nach Taxonomieniveau

Nachfolgend wird untersucht, ob sich Muster in den erreichten Punkten in Hinblick auf die Taxonomiestufen erkennen lassen.

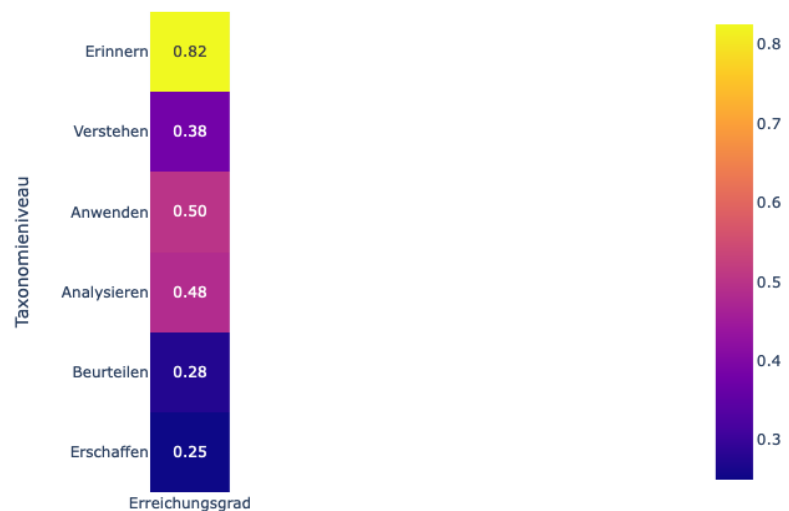


Abbildung 5 durchschnittliche Ergebnisse des Kontrolltests nach Taxonomiestufen

Zunächst soll allgemein die Erreichung der Lernziele betrachtet werden. Aus Abbildung 5 geht hervor, dass die durchschnittlichen Ergebnisse mit aufsteigendem Taxonomielevel geringer werden. Einzige Ausnahme ist zweite Niveau „verstehen“. Dies kann aber ein statistischer Ausreißer sein, der mit der geringen Anzahl an Teilnehmer oder konkreten Ausformulierung der Kontrollfragen zusammenhängt. Damit entspricht die generelle Erreichung der Lernziele in der Tendenz dem Erwarteten.

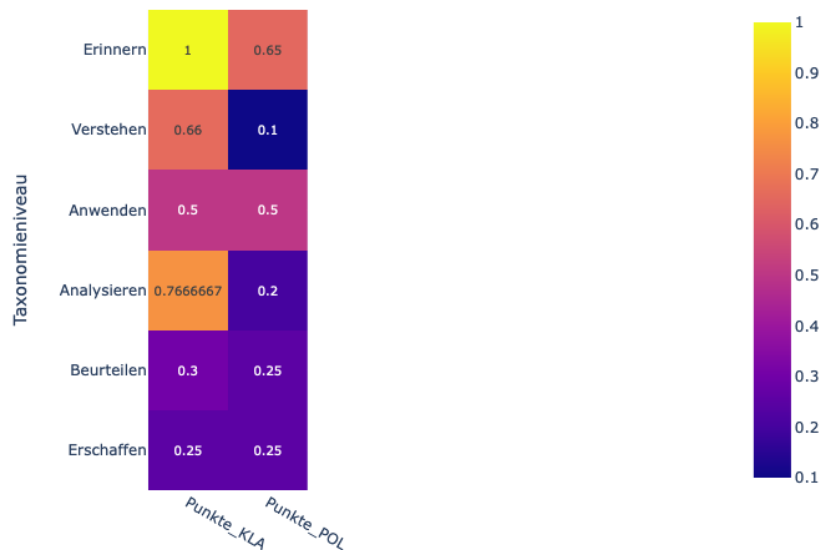


Abbildung 6 erreichte durchschnittliche Punkte nach Ansatz und Taxonomielevel

Gliedert man die Ergebnisse nach dem verwendeten didaktischen Ansatz auf, zeigt sich das in Abbildung 6 zu sehende Bild. Anhand dieser Abbildung kann man zunächst erkennen, dass in jeder Lernzielkategorie der Lernerfolg unter Verwendung klassischer Didaktik höher oder mindestens gleich hoch ist.

Betrachtet man die erreichten Punkte unter Verwendung klassischer Didaktik, so sinken diese mit aufsteigendem Taxonomieniveau. Ausnahme ist hier das Niveau „Analysieren“. Dies zeigt bei klassischer Didaktik außergewöhnlich gute Ergebnisse. Betrachtet man im Kontrast dazu die Punkte, die beim Ansatz problemorientiertes Lernen erzielt wurden, so zeigt sich hier ein heterogenes Bild. Die Gruppen „Erinnern“ und „Anwenden“ zeigen gute Ergebnisse, die übrigen Gruppen zeigen ein niedriges Erreichungsniveau.

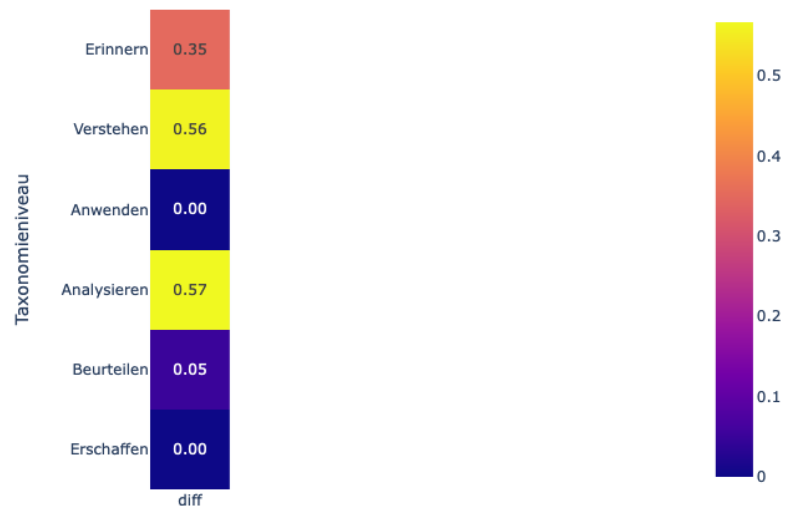


Abbildung 7 Differenz im Erreichungsgrad zwischen klassischer Didaktik und problemorientiertem Lernen

Abbildung 7 zeigt die Differenz im Erreichungsgrad der beiden didaktischen Ansätze. Wie bereits erwähnt zeigt die klassische Didaktik auf allen Taxonomieniveaus bessere oder gleiche Ergebnisse. In den Taxonomiestufen „Anwenden“, „Beurteilen“ und „Erschaffen“ zeigt sich wenig bis kein Unterschied. In den Stufen „Erinnern“, „Verstehen“ und „Analysieren“ führt die klassische Didaktik zu signifikant besseren Ergebnissen. Dieses Ergebnis kann derart interpretiert werden, dass die Stufen „Erinnern“, „Verstehend“ und „Analysieren“ eine Kenntnis von formalen Definitionen und Unterscheidungsmerkmalen von Konzepten voraussetzt. Diese Definitionen werden in der problemorientierten Didaktik weniger prominent vermittelt. Es kann angenommen werden, dass der erkennbare Unterschied auch aus diesem Umstand resultiert. Die übrigen Kategorien setzen eher die kreative Anwendung des Erlernten voraus. Eventuell ist dies ein Hinweis darauf, dass problemorientiertes Lernen in diesen Bereich einsetzbar ist.

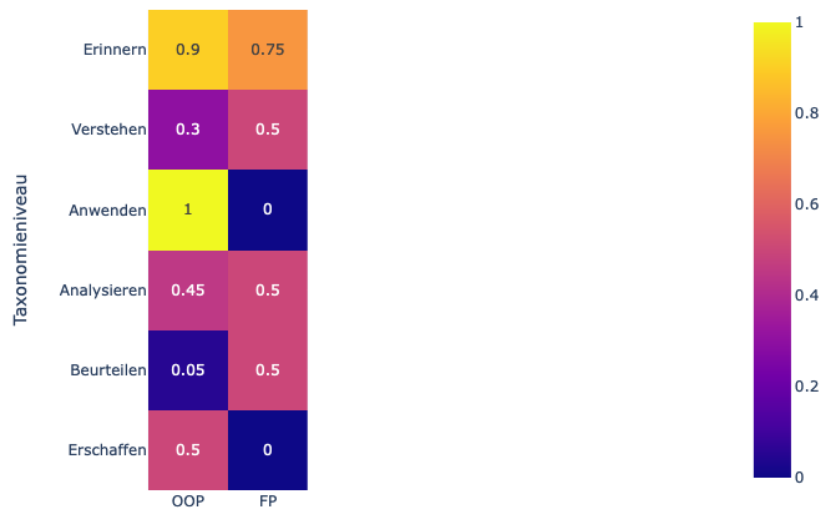


Abbildung 8 durchschnittlich erreichte Punkte nach Taxonomielevel und Lehrthema

Abschließend soll untersucht werden, wie Lernthema und Taxonomieniveau miteinander in Beziehung stehen. Dabei zeigt sich ein deutlich heterogenes Bild. In einigen Taxonomieniveaus sind die Erreichungsgrade eng beieinander, bei anderen liegt das eine oder andere Lernthema vorne.

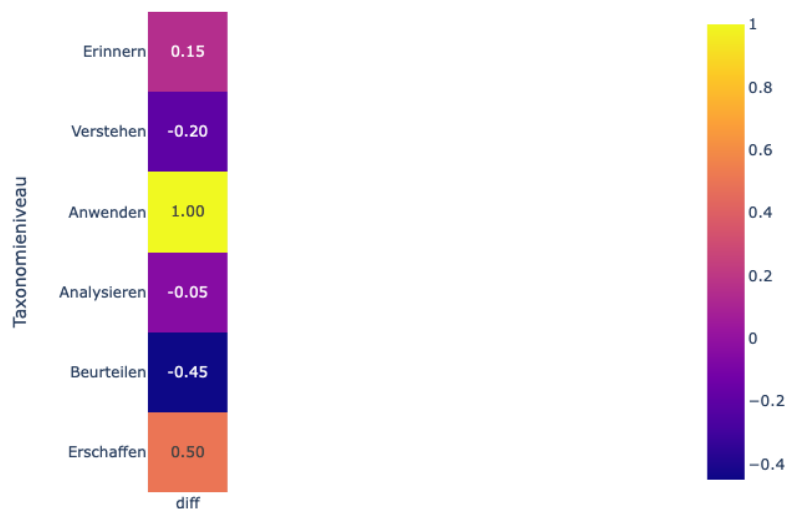


Abbildung 9 Differenz der durchschnittlich erreichten Punkte nach Taxonomieniveau und Lehrthema

Abbildung 9 zeigt die Differenz der Erreichungsgrade auf. Die erhobenen Daten lassen hier keine stichhaltigen Rückschlüsse zu. Die einzelnen Schwankungen können auf die speziellen Herausforderungen der Lernthemen zurückgeführt werden. Sie können aber auch aus der konkreten Ausgestaltung der Lerneinheiten oder Kontrollfragebögen resultieren.

### 5.3 Gesamtergebnis

Als zentrales Ergebnis des Versuches lässt sich festhalten, dass ein problemorientierter Lernansatz (zumindest in der hier verwendeten Form) in keiner relevanten Auswertung ein besseres Ergebnis erzielt als klassische Ansätze. Sowohl die Lernerfahrung der Teilnehmer (siehe Abs. 5.1.2) als auch die Ergebnisse der Kontrolltests (siehe Abs. 5.2.2) zeigen, dass hier ein klassischer didaktischer Ansatz bessere Ergebnisse erzielt. In 5.2.2 wurde dargelegt, dass der problemorientierte Ansatz konkurrenzfähige Ergebnisse in „kreativen“ Lernzielniveaus zeigt. Allerdings erreicht auch hier der problemorientierte Ansatz maximal das Niveau des klassischen Ansatzes.

Die Teilnehmer sagten aus, dass sie die Lernsituation im problemorientierten Ansatz als Prüfungssituation und damit als stressig empfunden haben. Eventuell muss hier die Lernsituation entsprechend angepasst werden, um hier eher ein Gefühl von Neugier und Entdeckerfreude zu vermitteln.

In 5.4 wird darauf eingegangen, dass die Ergebnisse dieses Versuches eine stark limitierte Aussagekraft haben. Daher sollte der Ansatz, problemorientierte Lehrmethoden in die Didaktik der Softwareentwicklung einzubinden, nicht komplett verworfen werden. Aus den Feedbackgesprächen mit den Teilnehmern geht hervor, dass diese sich zum Beispiel vorstellen können, die Grundlagen eines Themas klassisch zu vermitteln und die Kenntnisse problemorientiert zu vertiefen.

### 5.4 Kritische Reflektion

In Summe muss davon ausgegangen werden, dass die Ergebnisse dieses Versuches wenig aussagekräftig sind. Zunächst fußt die Auswertung lediglich auf zwei Teilnehmern. Die Auswahl geeigneter Teilnehmer hat sich in der Praxis schwierig gestaltet. Aufgrund der gewählten Beispielt Themen wurden Teilnehmer benötigt, die Grundlagen der Softwareentwicklung (Variablen, Schleifen, Bedingungen, etc.) beherrschen, darüber hinaus aber keine Kenntnisse haben. Dies grenzt die Personengruppe, in Vergleich zu Menschen ohne Vorkenntnisse oder Menschen mit fortgeschrittenen Kenntnissen im Bereich Softwareentwicklung stark ein (siehe: Abs. 4.3).

Zudem ist der Vergleich der beiden ausgewählten Themen funktionale und objektorientierte Softwareentwicklung im Versuch nur schwierig durchzuführen. Die konkreten gemessenen Lernzielergebnisse hängen stark vom Auswahl und Qualität der durchgeführten Unterweisungen sowie der Auswahl der Fragen in den Kontrolltests ab. Um den zeitlichen Aufwand in Grenzen zu halten, wurde pro Thema ein Kontrollfragebogen mit nur neun Fragen gestellt. Dies führt dazu, dass pro Taxonomieniveau lediglich ein bis zwei Fragen vorhanden sind. Ist eine dieser Fragen zu anspruchsvoll, missverständlich formuliert, etc. kann dies bereits zu einer starken Verfälschung der Ergebnisse führen. Zudem sind die Fragen in den Kontrollfragebögen grob nach Schwierigkeit und Thema geordnet. Nimmt man an, dass die meisten Teilnehmer den Kontrollfragebogen von vorne nach hinten ausfüllen, ist weiterhin anzunehmen, dass die späteren Fragen seltener beantwortet werden als die vorderen Fragen. Sind die Fragen nun nach Schwierigkeit und Niveau sortiert, kann dies zu einer Verzerrung bei Abprüfung der Kompetenzniveaus führen. Vereinfacht gesagt: Eine Frage, die das Niveau „Erschaffen“ abfragen soll, könnte eventuell vom Teilnehmer erfolgreich gelöst werden, wird aber in der Praxis nicht gelöst, da sie als letzte Frage im Fragebogen steht. Bei zukünftigen Versuchen sollte daher evaluiert werden, ob man Fragen stärker mischt und stärker darauf hinweist, dass die Fragen nicht in Reihenfolge beantwortet werden müssen.

Da die gleichen Kontrollfragebögen sowohl beim problemorientierten als auch beim klassischen Ansatz verwendet wurden, ist der Vergleich dieser Ansätze (in diesem Kontext) aber weiterhin stichhaltig.

Zudem muss die Ausgestaltung der Versuche hinterfragt werden. Aus den Feedbackgesprächen geht hervor, dass die angepeilten Lernziele als zu umfangreich für die Kürze der Versuche empfunden wurde. Dies zeigt sich auch bei der Untersuchung der Erreichungsgrade der Kontrollfragebögen (siehe Abs. 5.2.1). Für zukünftige Termine sollte der Lernumfang reduziert und/oder die Dauer der Unterweisungen verlängert werden.

Speziell letzteres ist in der Praxis schwer umzusetzen, ohne dass die Gruppe der Personen, die bereit sind an einem solchen Versuch teilzunehmen, verkleinern könnte.

Kritisch hinterfragt werden muss ebenfalls die Auswahl der vermittelten Themen. Es kann in Frage gestellt werden, ob Objektorientierte Programmierung und funktionale Programmierung ein vergleichbares Niveau an Vorkenntnissen voraussetzen.

Die Teilnehmer sagten dazu aus (siehe Abs. 5.1), dass objektorientierte Programmierung (OOP) als leichter und intuitiver zu verstehen wahrgenommen wurde. Zwar bringt OOP mehr neue Syntax und neue Konzepte mit als funktionale Programmierung (FP). Der große Vorteil ist aber, dass die Konzepte sich sehr gut mit Beispielen aus der realen Welt vermitteln und entsprechend optisch darstellen und Umsetzen lassen. Ruft man auf eine Variable „lamp1“ den Befehl „turnOn()“ auf und gleichzeitig schaltet sich eine Lampe im grafischen Interface an (siehe Anhang 20), ist dies auch ohne Vorkenntnisse zu verstehen.

Funktionale Programmierung hingegen bringt wenig neue Syntax mit. Lediglich Arrow Funktionen kommen neu hinzu und auch diese sind strenggenommen nur optional notwendig. Ansonsten fußt FP auf der Neuanwendung bekannter Konzepte, vor allem Funktionen. Zudem erfordert es eine starke abstrakte Denkfähigkeit. Die Konzepte und Ideen müssen in der Tiefe durchdrungen sein, bevor man selbst in der Lage ist, funktionale Programme zu schreiben. Dies ist für Teilnehmer, speziell wenn sie in den zu Grunde liegenden Konzepten (hier vor allem Funktionen) nicht absolut trittsicher sind, eine große Herausforderung. Dies wurde explizit von Teilnehmern in den Feedbackgesprächen genannt. Grundsätzlich kann aber festgehalten werden, dass der Versuch interessante Ergebnisse geliefert hat. Ein grundsätzliches Verwerfen der Ansätze ist nicht notwendig. Um wirklich belastbare Ergebnisse zu erzielen sollte der Versuchsaufbau jedoch kritisch hinterfragt und angepasst werden und vor allem ist eine sehr viel größere Teilnehmerbasis notwendig.

## 6 Literaturverzeichnis

- [1] R. Porsch, Einführung in die Allgemeine Didaktik, 2016.
- [2] „Lehre Laden - Ruhr Uni Bochum,“ [Online]. Available: <https://dbs-lin.ruhr-uni-bochum.de/lehreladen/planung-durchfuehrung-kompetenzorientierter-lehre/kompetenz-pruefen/lernzieltaxonomien/>. [Zugriff am 20 06 2022].
- [3] „Lehre Laden - Ruhr Uni Bochum - Vergleich der Lernzieltaxonomien,“ [Online]. Available: <https://dbs-lin.ruhr-uni-bochum.de/lehreladen/planung-durchfuehrung-kompetenzorientierter-lehre/lehr-und-lernziele/lernzieltaxonomien-im-vergleich/>. [Zugriff am 15 06 2022].
- [4] „educocktail - Universität für Weiterbildung Krems,“ [Online]. Available: <https://imbstudent.donau-uni.ac.at/educocktail/tag/anderson-krathwohl/>. [Zugriff am 20 06 2022].

## 7 Abbildungsverzeichnis

Abbildung 1 Darstellung der Lernzieltaxonomie nach Anderson und Krathwohl, Bild übernommen aus [2] .....	5
Abbildung 2 Erreichte Punkte, aufsummiert nach Thema.....	15
Abbildung 3 erreichte Punkte in den Kontrollfragebogen, aufsummiert nach Lernthema und didaktische Ansatz (eigene Darstellung).....	16
Abbildung 4 Punkteverteilung der Kontrollfragen nach didaktischem Ansatz (eigene Darstellung) .....	17
Abbildung 5 durchschnittliche Ergebnisse des Kontrolltests nach Taxonomiestufen .....	18
Abbildung 6 erreichte durchschnittliche Punkte nach Ansatz und Taxonomielevel .....	18
Abbildung 7 Differenz im Erreichungsgrad zwischen klassischer Didaktik und problemorientiertem Lernen.....	19
Abbildung 8 durchschnittlich erreichte Punkte nach Taxonomielevel und Lehrthema.....	20
Abbildung 9 Differenz der durchschnittlich erreichten Punkte nach Taxonomieniveau und Lehrthema .....	20



## 8 Tabellenverzeichnis

Tabelle 1 Kompetenzstufen nach Anderson und Krathwohl mit entsprechenden Operatoren	6
Tabelle 2 Lernziele objektorientierte Programmierung .....	10
Tabelle 3 Lernziele funktionale Programmierung .....	12

## 9 Anhang

- Anhang 1 Auswertung\_Feedbackgespräch;  
(30\_feedbackgespräche/Auswertung\_Feedbackgespräch.pdf)
- Anhang 2 Fragebogen\_TN2\_OOP;  
(40\_Auswertung\_Kontrollfragebogen/Fragebogen\_TN2\_OOP.pdf)
- Anhang 3 Auswertung\_Kontrollfragen;  
40\_Auswertung\_Kontrollfragebogen/Auswertung\_Kontrollfragen.xlsx)
- Anhang 4 Fragebogen\_TN2\_FP;  
(40\_Auswertung\_Kontrollfragebogen/Fragebogen\_TN2\_FP.pdf)
- Anhang 5 Fragebogen\_TN1\_OOP;  
(40\_Auswertung\_Kontrollfragebogen/Fragebogen\_TN1\_OOP.pdf)
- Anhang 6 Auswertung\_Fragebogen;  
(40\_Auswertung\_Kontrollfragebogen/Auswertung\_Fragebogen.ipynb)
- Anhang 7 Fragebogen\_TN1\_FP;  
(40\_Auswertung\_Kontrollfragebogen/Fragebogen\_TN1\_FP.pdf)
- Anhang 8 Präsentation\_FP; (20\_FP\_Didaktik/Präsentation\_FP.pdf)
- Anhang 9 Kontrollfragebogen\_material; (20\_FP\_Didaktik/Kontrollfragebogen\_material)
- Anhang 10 FP\_Umgebung\_problemlorientiertesLernen;  
(20\_FP\_Didaktik/FP\_Umgebung\_problemlorientiertesLernen)
- Anhang 11 Kontrollfragebogen\_FP; (20\_FP\_Didaktik/Kontrollfragebogen\_FP.pdf)
- Anhang 12 Aufgaben\_FP; (20\_FP\_Didaktik/Aufgaben\_FP.pdf)
- Anhang 13 FP\_klassische\_Didaktik\_material;  
(20\_FP\_Didaktik/FP\_klassische\_Didaktik\_material)
- Anhang 14 OOP\_klassische\_Didaktik\_Material;  
(10\_OOP\_Didaktik/OOP\_klassische\_Didaktik\_Material)
- Anhang 15 Aufgaben\_OOP; (10\_OOP\_Didaktik/Aufgaben\_OOP.pdf)
- Anhang 16 Kontrollfragebogen\_OOP; (10\_OOP\_Didaktik/Kontrollfragebogen\_OOP.pdf)
- Anhang 17 Präsentation\_OOP; (10\_OOP\_Didaktik/Präsentation\_OOP.pdf)
- Anhang 18 UML\_Diagramm.drawio; (10\_OOP\_Didaktik/UML\_Diagramm.drawio.pdf)
- Anhang 19 Kontrollfragebogen\_material;  
(10\_OOP\_Didaktik/Kontrollfragebogen\_material)
- Anhang 20 OOP\_Umgebung\_problemlorientiertesLernen;  
(10\_OOP\_Didaktik/OOP\_Umgebung\_problemlorientiertesLernen)