



Hochschule für
Wirtschaft und Recht Berlin
Berlin School of Economics and Law

Studienprojekt I

Schatzinsel

Entwickeln eines Spiels durch objektorientiertes Programmieren

fertiggestellt am TBD

•

Fachbereich Duales Studium Wirtschaft / Technik
Hochschule für Wirtschaft und Recht Berlin

Name: Jahn, Marko,
Name: Schenkewitz, Mario, 685593
Name: Zilius, Sven,
Fachrichtung: Duales Studium Wirtschaft • Technik
Studiengang: Informatik
Studienjahrgang: 2020
Betreuer HS: Zimmermann, Arthur
Anzahl der Wörter: TDB

Abstract

Kurzfassung

Inhaltsverzeichnis

Abstract	I
Kurzfassung	II
Inhaltsverzeichnis	IV
Abbildungsverzeichnis	V
Akronyme	V
1 Einleitung	1
2 Grundlagen	2
2.1 Objektorientierte Programmierung	2
2.2 Datenabstraktion und Verkapselung	2
2.3 Polymorphismus	3
2.4 Vererbung	3
2.5 Verwandte Konzepte	3
2.6 Auswahl des Programmiersprache und Umgebung	3
3 Realisierung	4
3.1 Verwendete Plattformen	5
3.1.1 Git und Github	5
3.1.2 Discord	5
3.2 Zusammenfassung der Anforderungen	5
3.3 Anforderung 1	5
3.3.1 Ansatz	5
3.3.2 Umsetzung	5
3.4 Anforderung 2	5
3.4.1 Ansatz	5
3.4.2 Umsetzung	5
3.5 Anforderung 3	5
3.5.1 Ansatz	5
3.5.2 Umsetzung	5
3.6 Anforderung 4	5
3.6.1 Ansatz	5
3.6.2 Umsetzung	5

4	Ergebnisse	6
5	Fazit	7
6	Ausblick	8
	Literaturverzeichnis	9
	Ehrenwörtliche Erklärung	10

Abbildungsverzeichnis

Akronyme

OOP Objektorientierte Programmierung

OOPL Objektorientierte Programmiersprachen

1 Einleitung

Die Objektorientierte Programmierung (OOP) ist ein heutzutage weit verbreitetes Computerprogrammiermodell oder Programmierparadigma. In der OOP werden Anwendungen um so genannte Objekte und deren Daten herum strukturiert und nicht um Funktionen oder Logik. So fokussiert sich die OOP auf Objekte mit denen eine Anwendung interagiert, diese Objekte können als Datenfeld mit Attributen und eigenem Verhalten angesehen werden.

Einen tieferen Einblick in das Thema der OOP soll der Abschnitt 2 bieten, in dem die Basis für die OOP näher erörtert wird.

Thema dieses Studienprojektes ist es sich dieses Programmierparadigma zu nutze zu machen um ein Spiel zu entwickeln.

Der Titel des Spiels ist „Schatzinsel“, es handelt von drei Piraten die sich auf einer Insel befinden, auf der ebenfalls ein Schatz versteckt ist. Es ist ihnen unbekannt wo sich dieser Schatz befinden. Durch zufällige Bewegung über die Insel wollen sie versuchen diesen Schatz zu finden. Jeder der Piraten hat verschiedene Eigenschaften, wie zum Beispiel Geschwindigkeit.

Sobald einer der Piraten den Schatz entdeckt, ruft er die anderen zu sich. Am Ende versammeln sie sich alle um den Schatz.

Dieser Umstand soll durch das Spiel modelliert und gegebenenfalls erweitert werden.

2 Grundlagen

2.1 Objektorientierte Programmierung

Objektorientierte Programmiersprachen (OOP) eignen sich durch die Konzentration auf die Objekte mit denen das Programm interagieren soll dazu, dass Software Wartbar und lesbar bleibt. Objekte und ihre Methoden können getrennt von einander erstellt und programmiert werden. Diese Unterteilung in Einzelteile ermöglicht des weiteren die Wiederverwendbarkeit von Code, also dem Quelltext, entweder in dem die Klasse selbst wiederverwendet wird, oder eine andere Klasse von ihr erbt. Dazu mehr im Abschnitt 2.4

OOP ist ein Programmierstil, der mittlerweile in Unternehmen, Industrie und akademischen Kreisen seinen Platz gefunden hat. Es wird ein natürlicheres und dadurch ein mächtigeres Entwurfparadigma insofern erwartet, dass ein Programmierer produktiver ist, wenn das Anwendungsmodell näher am Problem modelliert werden kann, dass gelöst wird.

Anders gesagt ist eine OOP mächtiger, weil es einem Programmierer erlaubt ein Problem zu lösen, dass die Struktur einer Welt nachbildet in welcher das Problem entstand. Vertrautheit mit solch einer Sprache eröffnet Wege zu einer klaren und natürlichen Lösung. [Yae14]

Es werden also im betrachteten Fall der Schatzinsel die Objekte, die aus der echten Welt bekannt sind, so modelliert, dass die Interaktionen und Verhaltensweisen natürlich in einer Anwendung modelliert werden können.

2.2 Datenabstraktion und Verkapselung

Einer der ersten Schritte bei der OOP ist es alle Objekte zu sammeln, die manipuliert werden müssen. Im Fall vom Spiel „Schatzinsel“ gibt es zum Beispiel Piraten, eine Insel, ein Schatz, aber auch eine graphische Benutzeroberfläche die alle als Objekt betrachtet werden können. Sie müssen untereinander aber auch mit dem Benutzer interagieren können.

Einer der großen Durchbrüche bei der Gestaltung von Programmiersprachen geschah, als es ermöglicht wurde, dass Programmierer eigene Datentypen definieren

konnten. C.A.R. Hoare's record class in Simula67 gilt als die Quelle der Idee. [Yae14] Verbreitung der durch Programmierer definierten fand durch die Sprache Pascal statt. [Wir71]

So kann in Pascal ein Verbund definiert werden, der selbst aus Daten besteht.

```
1 TYPE Auto = RECORD
2   KFZ: String[12];
3   Gewicht: Real;
4   AnzPassagiere: Integer;
5 END;
```

Listing 2.1: Verbund Beispiel Definition [Zim20]

Dieses Beispiel definiert einen neuen Datentyp *Auto*, in dem jeweils die Art des KFZ, das Gewicht und die Anzahl der Passagiere angegeben wird. Nun kann man den Typen *Auto* nutzen um beliebig viele Daten-Entitäten zu erstellen.

```
1 VAR a : Auto;
2 BEGIN
3   a.KFZ := 'B XY 123456';
4   a.Gewicht := 555.77;
5   a.AnzPassagiere := 5;
6 END;
```

Listing 2.2: Verbund Beispiel Instantiierung [Zim20]

Manipulationen der Daten innerhalb des Verbunds werden extern vom Datentyp selbst betrachtet, die Syntax der Sprache indiziert somit nur eine Gruppierung von Daten.

Verbunde in Pascal werden homogen genannt, wenn alle Daten innerhalb des Verbunds den selben Typ haben. Sie sind heterogen, wenn es verschiedene Typen, wie im Beispiel, gibt.

2.3 Polymorphismus

2.4 Vererbung

2.5 Verwandte Konzepte

2.6 Auswahl der Programmiersprache und Umgebung

3 Realisierung

3.1 Verwendete Plattformen

3.1.1 Git und Github

3.1.2 Discord

3.2 Zusammenfassung der Anforderungen

3.3 Anforderung 1

3.3.1 Ansatz

3.3.2 Umsetzung

3.4 Anforderung 2

3.4.1 Ansatz

3.4.2 Umsetzung

3.5 Anforderung 3

3.5.1 Ansatz

3.5.2 Umsetzung

3.6 Anforderung 4

3.6.1 Ansatz

3.6.2 Umsetzung

4 Ergebnisse

5 Fazit

6 Ausblick

Literaturverzeichnis

- [Wir71] Wirth, N. (1971). *The Programming Language Pascal*. Acta Informatica 1. pp. 35–63.
- [Yae14] Yaeger, D. P. (2014). *Object-Oriented Programming Languages And Event-Driven Programming*. Mercury Learning and Information, Dulles, Virginia; Boston, Massachusetts; New Delhi.
- [Zim20] Zimmermann, A. (2020). Einführung in die Programmierung, Aggregierte Datentypen Übung. <http://azdom.de/hwr/prog/>.

Ehrenwörtliche Erklärung

Ich erkläre ehrenwörtlich:

1. dass ich meine Bachelor-Thesis selbstständig verfasst habe,
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe,
3. dass ich meine Bachelor-Thesis bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Ort, Datum

Jahn, Marko

Ort, Datum

Schenkewitz, Mario

Ort, Datum

Zilius, Sven