

Ruby als Lernprogrammiersprache

Nicolai Böttger und Jon-Steven Streller

Seminar-Arbeit im Studiengang „Angewandte Informatik“

15. Juni 2019



Autor 1: Nicolai Böttger
1476431
nicolai.boettger@stud.hs-hannover.de
Verfasste Seiten/Abschnitte: ...

Autorin 2: Jon-Steven Streller
1475759
steven.streller@stud.hs-hannover.de
Verfasste Seiten/Abschnitte: ...

Prüferin: Prof. Dr. Dennis Allerkamp
Abteilung Informatik, Fakultät IV
Hochschule Hannover
dennis.allerkamp@hs-hannover.de

Selbständigkeitserklärung

Mit der Abgabe der Ausarbeitung erklären wir, dass wir die eingereichte Seminar-Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die von uns angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht haben.

Hannover, den 15. Juni 2019

Inhaltsverzeichnis

1 Einführung

Die vorliegende Arbeit entstand im Rahmen des Seminars „*Programmiersprachen für den Einstieg*“. Ziel des Seminars war es mehrere Programmiersprachen auf ihre Tauglichkeit im Anwendungsbereich einer pädagogischen Lehrveranstaltung zu analysieren. Hierbei wurde jede Programmiersprache einzeln von verschiedenen Studierenden in Zweiergruppen vorgestellt und die jeweiligen Ergebnisse Ihrer Forschung präsentiert. Nachfolgend entstand ein Austausch zwischen den Studierenden über positive und auch negative Auffälligkeiten in der Präsentation.

1.1 Vorgehensweise

Um eine sinnvolle Bewertung der Sprache in Bezug zum Thema des Seminars zu erreichen haben wir zuerst nach Studien gesucht, welche Kriterien definieren, nach denen man eine Programmiersprache bewerten kann. Danach haben wir angefangen uns selbst in unsere vorzustellende Sprache „*Ruby*“ einzuarbeiten. Nachdem wir ein grundsätzliches Verständnis der Sprache bekommen haben, haben wir angefangen „*Ruby*“ in Bezug zu den vorher gewählten Kriterien zu bewerten.

1.2 Aufbau der Arbeit

Im folgenden **Hauptteil** wird die Programmiersprache „*Ruby*“ vorgestellt. Hierbei werden auf besondere Merkmale der Sprache eingegangen und *Anwendungsbeispiele* visualisiert. Anschließend werden die Methodik und die Kriterien erläutert und definiert. Zum Abschluss werden die Forschungsergebnisse in Form einer Bewertung dargelegt.

2 Hauptteil

2.1 Was ist Ruby?

Im Jahr 1995 veröffentlichte *Yukihiro Matsumoto* die erste Version (Version 0.95) von „*Ruby*“. *Yukihiro Matsumoto* lies sich während der Entwicklung von „*Ruby*“ von mehreren Programmiersprachen wie z.B. Perl, Smalltalk, Lisp, Ada inspirieren. *Ruby* besitzt eine tief integrierte Objektorientierung¹ (kurz OO).

¹ Ein System besteht in der Objektorientierung ausschließlich aus Objekten, die miteinander über Nachrichten kommunizieren. Jedes Objekt verfügt über Eigenschaften und Methoden. Die Eigenschaften beschreiben dabei über ihre Werte den Zustand eines Objektes, die Methoden die möglichen Handlungen eines Objektes. vgl. [WL01]

2.2 Arbeitsweise von Ruby

An dieser Stelle wird kurz auf die interne Arbeitsweise von Ruby eingegangen. Ruby-Code wird ein Statement nach dem anderen, von einem Interpreter übersetzt und ausgeführt. Im Gegensatz zum Compiler, analysiert dieser den Code schneller und das Schrittweise übersetzen erleichtert das Finden von Fehlern im Code, allerdings hat ein kompiliertes Programm meist eine bessere Performance.

2.3 Methodik

Um die Programmiersprache Ruby in Bezug zur Fragestellung, ob sie sich als Programmiersprache für den Einstieg eignet bewerten zu können, wurde als Methodik ein Kriterienkatalog erstellt, welcher die Sprache in hinsichtlich verschiedener Bereiche untersucht. Dieser Katalog besteht aus den Kriterien Einstiegsfreundlichkeit, Skalierbarkeit, Verständlichkeit, Dokumentation, Verbreitung der Sprache und Ausstattung der Schule. Eine besondere Gewichtung wurde hierbei auf die Kriterien Skalierbarkeit und Verständlichkeit gelegt, da diese die grundlegenden Aspekte zum Verstehen der Sprache sind. Nach einer kurzen Erläuterung der Kriterien, werden diese in Bezug zur Programmiersprache Ruby gesetzt und es wird erörtert, inwiefern das Kriterium zutrifft (trifft voll zu, trifft teilweise zu, trifft nicht zu)

2.3.1 Einstiegsfreundlichkeit

Das Kriterium der „Einstiegsfreundlichkeit“ fasst alles zusammen, was benötigt wird um mit dem Programmieren in der Sprache anzufangen. So sollte die Installation von möglichen Compilern oder Interpretern nicht zu aufwändig und ohne große Fachkenntnisse vorgenommen werden können.

2.3.2 Skalierbarkeit

3 Schlussbemerkungen

4 Literaturverzeichnis

[WL01] Wirtschaftslexikon Gabler