

# Auswertung der Evaluation zu „Programmieren in Rust“

2016/2017, Universität Osnabrück

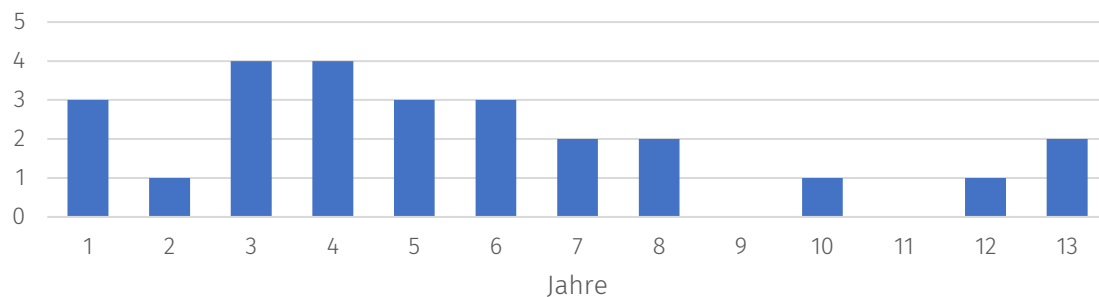
Dozent: Lukas Kalbertodt

Anzahl Teilnehmer: 25

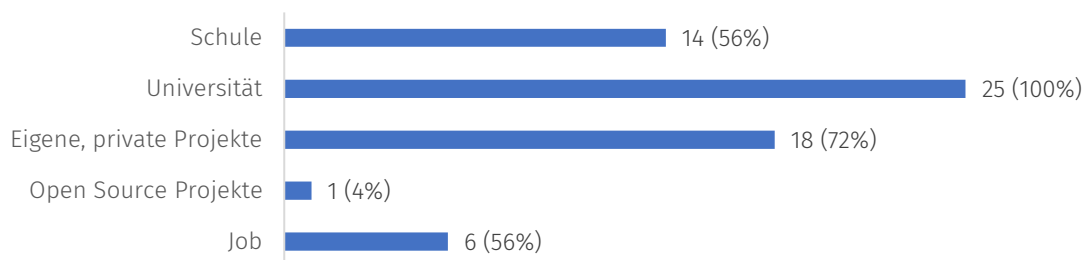
## Abschnitt: „Über dich“

---

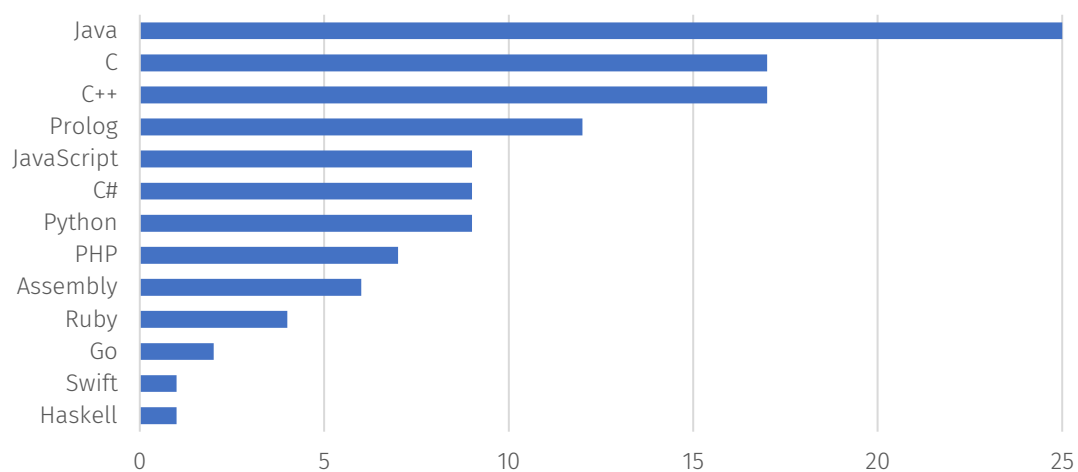
Seit wie vielen Jahren programmierst du bereits?



Wofür hast du bereits programmiert?

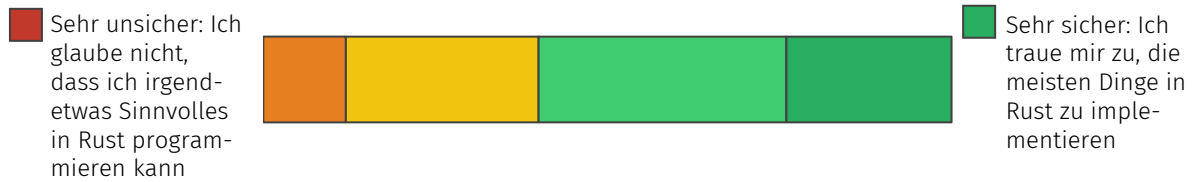


Mit welchen Sprachen bist du vertraut?



## Abschnitt: „Die Sprache Rust“

Wie sicher fühlst du dich inzwischen in der Rust-Programmierung?



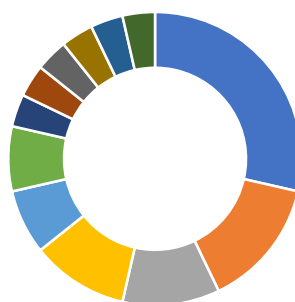
Wie viel Spaß macht dir die Programmierung in Rust (verglichen mit anderen Sprachen)?



Wie neu waren die Konzepte von Rust für dich?



Was war das überraschendste/neueste Konzept für dich?

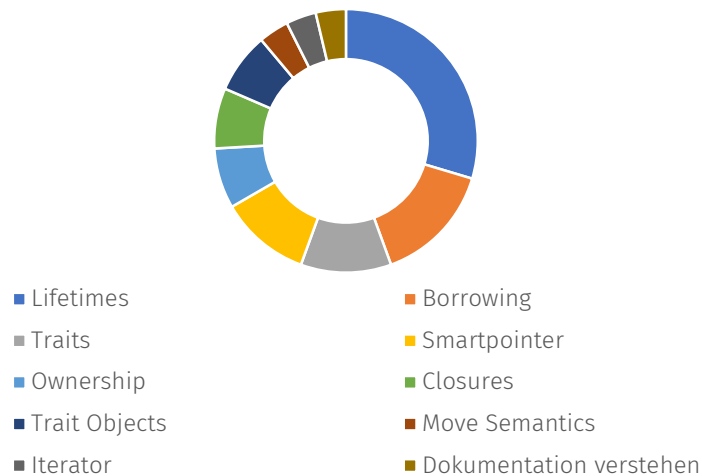


- |                      |                        |                           |
|----------------------|------------------------|---------------------------|
| ■ Ownership-System   | ■ Lifetimes            | ■ Iteratoren              |
| ■ Macros             | ■ Explicit mutability  | ■ Closures                |
| ■ Trait Objects      | ■ Traits               | ■ Low Level Kram          |
| ■ Speichersicherheit | ■ Abwesenheit von null | ■ Rust's Fehlerbehandlung |

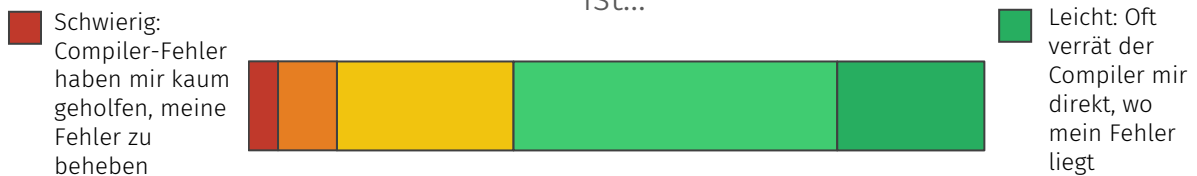
Wie schwierig ist es deiner Meinung nach, Rust zu lernen?



Welcher Teil von Rust war am schwierigsten zu erlernen?



Debugging mit Hilfe des Compilers (also ohne Internet) ist...



Im Vergleich zu anderen Sprachen: Bemerkt ihr mehr Fehler eures Programms zur Kompilierzeit oder zur Laufzeit?



## Als wie robust empfindest du deine in Rust geschriebenen Programme (im Vergleich zu anderen Sprachen)?



## Im Internet findet man zu (Problemen mit) Rust ...



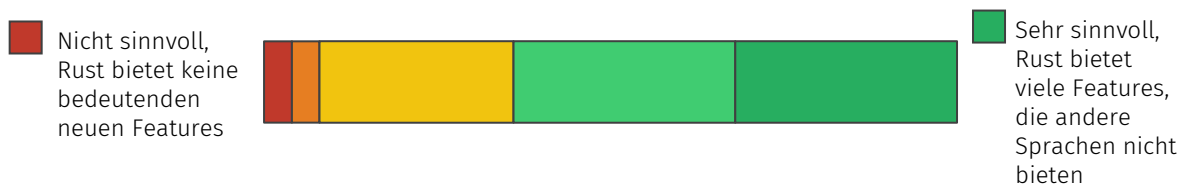
## Als wie kompakt würdest du Rust beschreiben?



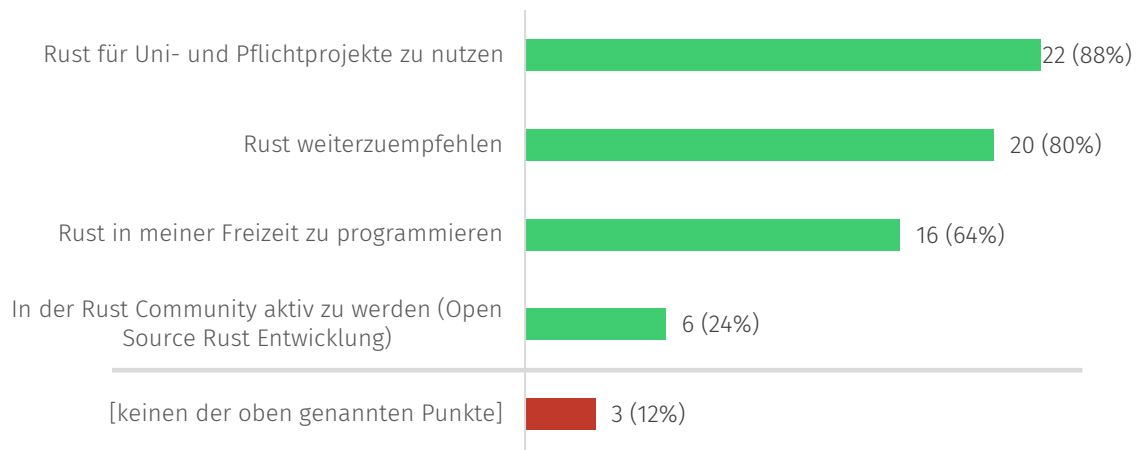
## Als wie elegant würdest du Rust beschreiben?



## Wie sinnvoll ist Rust's Existenz/Entwicklung für dich?



## In der Zukunft könnte ich mir vorstellen, ...



### Kommentare zu Rust:

Typkonstruktoren wären nett.

Die Schwierigkeit der Lernens hängt meiner Ansicht nach auch etwas mit den Vorkenntnissen in Bereich C/C++ zusammen (z.B. Pointer-System). Ich hab bisher nur an einigen wenigen Stellen "mutable statics" vermisst. Diese ließen sich jedoch in den meisten Fällen vermeiden.

Rust fehlt definitiv noch eine eigene GUI und möglicherweise eine eigene Entwicklungsumgebung

So gut wie alles, was in Info B megascheiße war, ist in Rust gar nicht mehr soooo furchtbar.

Wenn es möglich ist die ganzen Funktionen zu Iteratoren, Options und so weiter die es ermöglichen mehrere Zeilen in eine zu verpacken

Rust fehlt Zeit und gut dokumentierte Crates und bessere Doku für Traits.

Gute Elemente: match, Option, Result (letztere können aber auch nervig werden)

Rust versucht, alles in einer Sprache zu vereinen - das kann problematisch werden.

Nervigste Seite: schlecht dokumentierte Traits und die aufwendige Einrichtung.

Hin und wieder gehen Programme tatsächlich mal mysteriöserweise kaputt, ganz gefeilt ist Rust da natürlich auch nicht gegen, aber im Vergleich zu anderen Sprachen deutlich robuster. Debuggen von Runtime-Fehlern ist in Rust ähnlich schwer wie in C++, vielleicht wäre es sinnvoll gewesen mal auf Debugging-Tools wie den gdb einzugehen. Hin und wieder kommt auch einmal eine irreführende Fehlermeldung des Compilers (generell weiß man, dass man ein Problem hat, wenn in der Fehlermeldung eine Issue auf Github referenziert wird). Theoretisch bietet Rust in vielen Fällen die Möglichkeit für sehr viel kompakteren Code als in anderen Sprachen, aber wenn man beispielsweise mal den größten Wert aus einem Float-Vector sucht (von dem man bereits weiß, dass er in dieser Anwendung niemals NaN enthalten kann) wird es meist deutlich komplexer als in anderen Sprachen. Natürlich hat das gute Gründe, der Fakt bleibt aber.

Rust fehlt gut dokumentierte Crates,

Schön, dass es auf vielen Betriebssystemen leicht zu installieren/verwenden ist und cargo ist sehr praktisch. Am nervigsten: aussichtslose Verzweigung bei manchen Problemen und dass einige Probleme (z.B. mit Mutability und mehreren Referenzen) nur mit Expertenwissen sehr umständlich gelöst werden können Was fehlt: Vererbung, ein schöneres/leichter verwendbares Klassensystem.

Rust fehlt Vererbung von Structs

## Abschnitt: „Die Vorlesung“

Wie verständlich war die Vorlesung?



Wie verständlich waren die Folien?



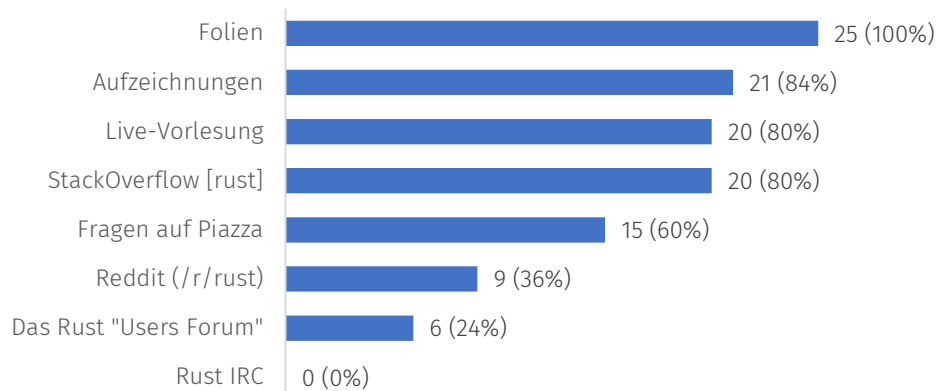
Ich habe das Gefühl, die Vorlesung hat mir sinnvolles & anwendbares Wissen vermittelt



Der Arbeitsaufwand des Kurses war...



Ich habe Folgendes in Anspruch genommen



## Kommentare zur Vorlesung:

Der Umfang der Übungsaufgaben variierte teilweise etwas stark, was die Zeitplanung der Woche erschwerte.

Vorbereitung auf zu stellende Fragen während des Testats hätte besser sein können.

Ansonsten top Veranstaltung, hat viel Spaß gemacht und mein Interesse an Rust sehr gefördert.

---

Die Vorlesung war die Beste, an der ich bisher teilgenommen habe!

---

Der Tutor ist super, schönes Info-A-Déjà vu ^.^

---

Ich denke ein Punktesystem bei den Testaten und umfangreichere verbale Rückmeldung zur Leistung der Studenten wäre gut.

---

Das System, die Aufgaben auf github abzugeben war extrem gut, das sollte bei jeder Vorlesung so gemacht werden.

---

Die Vorlesung war inhaltlich sowie didaktisch 1a, besser geht's nicht!

---

Die Folien sind leider nur in Verbindung mit der Vorlesung interessant. Auch wenn ich eine Vorlesung gesehen habe ist es nur bedingt möglich in den Folien nochmal eine Sache nach zu lesen. Meist musste ich die Vorlesung nochmal einschalten. Zudem waren die Links nicht nutzbar.(Habe ich aber persönlich auch nicht genutzt.

---

Tip top!

---

Mit eine meiner Lieblingsvorlesungen

---

Es war die beste Vorlesung, die ich bis jetzt gehört habe. Durch deine Begeisterung machst du es leicht, dir zu folgen und aufmerksam zu sein. Die Testate waren auch schön locker und man konnte nochmal Fragen stellen, wenn man etwas nicht richtig verstanden hatte, ohne gleich befürchten zu müssen, dann durchzufallen.

Der mehr oder weniger Zwang die Reviewkommentare auch wirklich einzuarbeiten war störend. Das Review war gut und wichtig, aber die Pflicht den Code daran anlehnd zu verbessern hat häufig nur dazu geführt, dass ich die Sachen aus der Musterlösung kopiert habe. Die Kenntnisnahme der Kommentare ist wichtig, aber sollte in der Eigenverantwortlichkeit der Leute liegen. Wenn manche Schwachstellen beim nächsten mal nicht besser werden, kann man denjenigen noch mal darauf hinweisen oder halt mal durchfallen lassen.

Bis Weihnachten haben VL + Aufgaben Spaß gemacht. Danach leider nicht mehr so, da es zu Low-Level wurde. Statt versuchen alle Konzepte, die Rust zu bieten hat, auf ein paar Folien zu packen und diese alle einmal kurz anzusprechen (X Konzepte pro VL) lieber etwas weglassen und dafür mehr Zeit mit bestimmten Themen zubringen. Jetzt hat man zwar von vielem mal gehört, aber die praktische Umsetzung und in welchen Situationen man es überhaupt gut verwenden könnte ist eine andere Sache. Lieber mehr Programmierpraxis mit anderen Konzepten aufbauen, sodass man sich am Ende wirklich in der Lage fühlt etwas in Rust umsetzen zu können. Den Rest kann man sich dann bei Bedarf selbst aneignen.

Das Projekt war soweit gut zur Einübung, aber es hängt halt stark davon ab, was für ein Projekt gewählt wird, ich denke unser Projekt war für 40h pro Person zu groß. Bei einigen Gruppen hatte es auch den Anschein, dass dort deutlich mehr als die Zeit investiert wurde - alles schön und gut, aber dadurch werden die Ergebnisse nicht mehr wirklich bewertbar. Besser wäre es wohl gewesen zu sagen, ihr habt jetzt 1,5 Wochen und dann ist es vorbei, dadurch wäre die Diskrepanz in den Ergebnissen deutlich geringer. Die Leute, die viel Zeit und Lust in den 4 Wochen hatten, konnten ordentlich was schaffen und andere entsprechend weniger.

---

z.B. Lifetimes (wenigstens die Grundlagen) hätte ich zu einem früheren Zeitpunkt in der Vorlesung sinnvoll gefunden, da ich vorher oft damit Probleme bei den Aufgaben hatte. Der Zeitraum des Projektes (während der Klausurenphase) war sehr schlecht gewählt. Sehr viel Aufwand für wenig ECTS.

Es war schön, dass manche Aufgaben interessant waren, sodass es Spaß gemacht hat daran zu arbeiten. Manchmal waren die Testat-Fragen etwas fies, aber es war sehr gut, dass Jonas sich immer Zeit genommen hat schwierige Konzepte nochmal zu erklären und Fragen zu beantworten.

Ich würde mir mehr Kurse wünschen, die moderne Themen wie Rust behandeln. Sehr gute Vorlesung, sehr motivierter Dozent und sehr gute Tutoren, sehr guter Vorlesungsstil.

Das System der Übungen mit Github und vor allem die Reviews der pull requests nach Abgabe der Aufgaben war unschlagbar, das würde ich mir so in allen Veranstaltungen wünschen da man damit wirklich das lernt, was man lernen sollte, nämlich guten und eleganten Code zu schreiben anstatt irgend eine Lösung zu hacken. Danke an die Tutoren.

Man könnte auch darüber streiten ob so viel Assembly und Optimierung in einem Rust-Kurs gerechtfertigt sind. Da das aber in keinem anderen Kurs (den ich bisher besucht habe) wirklich mit praktischem Aspekt zum Pflicht-Programm gehört hat, war es wohl gut, das mal gemacht zu haben.

Wirklich furchtbar war nur die Wahl des Zeitraums für das Praktikum, selbst mit vier Wochen, in denen man theoretisch hätte arbeiten können, haben die verschiedenen Klausuren dafür gesorgt, dass man selber mitunter kaum Zeit für die Bearbeitung hatte. Vor allem hat das darin resultiert, dass es kaum einen Zeitraum gab, in der mal mehr als die Hälfte der Gruppe zusammenarbeiten konnte. Das gekoppelt mit dem enormen Umfang der Aufgaben des Praktikums (viele der Projekte sind kaum innerhalb einer Woche zu schaffen, alleine das Raussuchen und Einarbeiten in benötigte Crates hat mitunter länger gedauert) war ein echtes Problem.

Das war aber auch der einzige wirkliche Wermutstropfen. Super Veranstaltung, danke dafür! (Schade dass künftige Jahrgänge diesen Kurs wohl nicht mehr besuchen können...)

---

Extrem gute Veranstaltung. Sehr gut Strukturiert, sehr gute Slides, sehr gute Vorlesungen, interessante und relevante Übungsaufgaben, sehr gute Tutoren. Sehr gutes Übungssystem mit dem verpflichtenden verbessern der abgegebenen Aufgaben. Abschlussprojekt für Programmierkurs optimal.

Habe mich immer auf die nächste Vorlesung gefreut. Also so stelle ich mir eine gute Veranstaltung vor. Wenn alle Kurse mit solcher Qualität geführt würden, hätte ich vermutlich schon längst eine Konkurrenz zu Google aufgebaut und wäre Milliardär.^^

---

Cooler Vorlesung, finde es klasse, dass es aufgezeichnet wird.

Hi Lukas, ich wollte nur mal eben eine Fanmail abgeben. Ich hab die Vorlesung eigentlich eher durch Zufall besucht und ich bin absoluter Fan der ersten Stunde. Die Vorlesung ist super strukturiert, der Inhalt wird durch das Liveprogrammieren richtig gut unterstützt und aufgelockert und auch die Vortragsweise und das Einbringen der Hörer durch häufige kleine Fragen ist einfach ideal um mitzukommen und viele Einblicke zu kriegen. Ich muss gestehen, dass ich mich als nicht sonderlich guten und innovativen Programmierer sehe und das ist auch nach diesem Kurs noch nicht so. Aber trotzdem hat mir der Kurs so geholfen, mich im Programmieren etwas sicherer zu fühlen. Auch die komplizierteren Themen waren meistens irgendwann einigermaßen verständlich und die Anwendung ging dann auch immer irgendwie, selbst wenn ich immer noch nicht ganz dahinter gekommen bin, warum manche Sachen jetzt funktionieren und warum nicht. Ich fühle mich definitiv in die (zugegebenermaßen recht sympatische) Rust-Sekte rekrutiert und kann mir gut vorstellen, zukünftig auch andere Probleme mit Rust zu lösen. Danke für den tollen Kurs und schade, dass es ihn in Zukunft nicht mehr geben wird, denn die Vorlesung ist definitiv eine der besten die, die ich je besucht habe. <3

---

Danke für das extrem schöne Feedback!

