

252-0027

Einführung in die Programmierung

Einführung

Manuela Fischer, Malte Schwerhoff

Departement Informatik

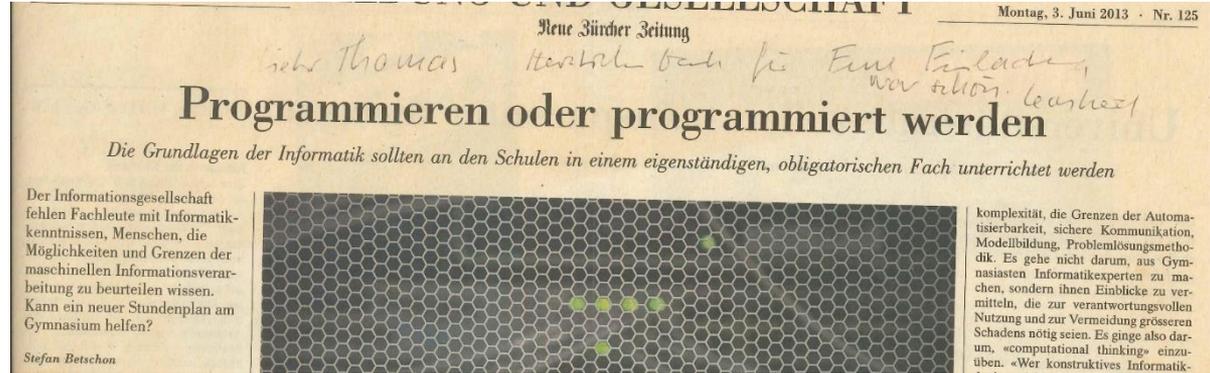
ETH Zürich

«Educators, generals, dieticians, psychologists, and parents program. Armies, students, and some societies are programmed.»

Alan Perlis (Foreword to «Structure and Interpretation of Computer Programs»,
H. Abelson and G. J. Sussman, 1985)

Für uns: Computer Programming

Vor 5+ Jahren



Süddeutsche.de Karriere

25. April 2017, 16:36 Tech-Jobs

Wer nicht programmiert, muss putzen!

THE HILL

31. Dez. 2019

Enrichment > Education

Biden tells coal miners to “learn to code”

By Alexandra Kelley | Dec. 31, 2019

Seit Kurzem...

COMMUNICATIONS OF THE ACM

HOME | CURRENT ISSUE | NEW!

[Home](#) / [Magazine Archive](#) / [January 2023 \(Vol. 66, No. 1\)](#) / [The End of Pr](#)

VIEWPOINT

The End of Programming

By Matt Welsh

Communications of the ACM, January 2023, Vol. 66 No. 1, Pages 34-35

RESEARCH-ARTICLE OPEN ACCESS 



Programming Is Hard - Or at Least It Used to Be: Educational Opportunities and Challenges of AI Code Generation

Authors:  [Brett A. Becker](#),  [Paul Denny](#),  [James Finnie-Ansley](#),  [Andrew Luxton-Reilly](#),  [James Prather](#),
 [Eddie Antonio Santos](#) [Authors Info & Claims](#)

SIGCSE 2023: Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 • March 2023 •
Pages 500–506 • <https://doi.org/10.1145/3545945.3569759>



Large Language Models and the End of Programming

Date & Time	May 9, 2023 12:00 PM Eastern Time (US and Canada)
Webinar ID	942 7707 2999

(Soll man) Programmieren lernen?

- Programmieren – das Latein des 21. Jahrhunderts?
- stärkt logisches Denken, aber nicht wirklich wichtig
 - Github Copilot macht die Arbeit für uns

Regelunterricht > Fächer > Latein

Latein



Open Source Product

Introducing GitHub Copilot: your AI pair programmer

Today, we're launching a technical preview of GitHub Copilot, a new AI pair programmer that helps you write better code.



GitHub
Copilot

- “Programmieren” mit GitHub Copilot
 - **Spezifikation** (Aufgabenstellung) eingeben
 - Copilot liefert ein (hoffentlich?!?) korrektes Programm

Count "coXe": Spezifikation

```
5 // Return the number of times that the
6 // string "code" appears anywhere in the
7 // given string, except we'll accept any
8 // letter for the 'd', so "cope" and "cooe"
9 // count.
```

Count "coXe": Vorschlag für Name

```
5 // Return the number of times that the
6 // string "code" appears anywhere in the
7 // given string, except we'll accept any
8 // letter for the 'd', so "cope" and "cooe"
9 // count.
10 // countCode("aaacodebbb") → 1
```

Count "coXe": 1. "Lösung"

```
// count.
```

```
public static int countCode("aaacodebbb")
```

```
public static int countCode("aaacodebbb"){  
    int count = 0;  
    for(int i = 0; i < str.length()-3; i++){  
        if(str.substring(i, i+2).equals("co")  
            count++;  
        }  
    }  
    return count;
```

Count "coXe": Lösung

```
// Return the number of times that the
// string "code" appears anywhere in the
// given string, except we'll accept any
// letter for the 'd', so "cope" and "cooe"
// count.
public static int countCode(String str){
    int count = 0;
    for(int i = 0; i < str.length() - 3; i++){
        if(str.substring(i, i + 2).equals("co") &&
            str.charAt(i + 3) == 'e'){
            count++;
        }
    }
    return count;
}
```

Count "coXe" mit Copilot: Fazit

- Wählte guten Namen für das Programm
- Brauchte (etwas) Hilfe mit dem Parameter



Copilot: Beobachtung

- Copilot “lernte” mit öffentlichen Programmieraufgaben
 - kennt beliebte Aufgaben (Klassiker)
 - kennt nicht nur die Lösungen...
 - ... sondern auch das nächste Problem

“ As far as I can tell,
Copilot was
specifically trained
on all the intro
programming
assignments ever

E. Berger, Coping w/ Copilot

Copilot macht EProg-Übungen (2018)

im 1. Versuch

Woche	Thema	Punkte
4	String-Addition: loops, arrays	0
5	Tool Rental: classes, arrays, iteration, JUnit	0
6	Valleys & Hills: arrays, data analysis, I/O	0
7	String Interleaving: recursion	0
8	List Reversal: references, working w/ classes	0
9	Class Puzzle: inheritance (<i>w/o classes or driver</i>)	0
10	Desk Calculator: inheritance, recursion	0
11	Data Analysis (FIFA): ArrayList<..>, Map<..>	0
12	Sublist Palindrome: Set<List<..>>, exceptions	100

Your AI pair programmer

Copilot mit Hinweisen

Woche	Hinweise	Punkte
4	4	100
5	3	100
6	4	100
7	2	0
8	1	100
9	12	100
10	5	100
11	3	100
12	0	100

2018

Woche	Hinweise	Punkte
4	1	100
5	2	25
6	2	67
7	16	100
8	5	75
9	10	60
10	4	100
11	4	100
12	7	32

2021

Copilot: Your Pair Programmer

- Erwartet Hinweise, wie zum Beispiel
 - Einfügen von Variablen
 - Änderung des Kontrollflusses
 - Hilfe bei Implementation einer Datenstruktur
 - Attribute und Methoden von Objekten
 - Definition von Vergleich(sprädikat)en
- Um sinnvolle Hinweise geben zu können, müssen Sie programmieren können!
- Copilot ist Ergänzung, kein Ersatz!



Fragen dazu?
Dann sind Sie hier
richtig!

Programmieren

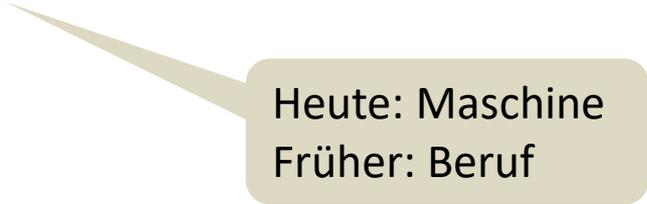
- **Programmieren**: Erstellen von Programmen
- **Programm**: eine Folge von Anweisungen an Computer
 - *prógramma* (griechisch): Bekanntmachung; Tagesordnung
 - Verfasst in einer Programmiersprache, für Menschen lesbar
 - Programm realisiert einen Algorithmus
- **Algorithmus**: Schritt-für-Schritt-Anleitung fürs Lösen eines Problems
 - Ausführlich diskutiert in der A&D-Vorlesung

Wie programmieren lernen?

- Sprache des Computers lernen
- Wie bei Fremdsprache
 - Zuerst Vokabeln...
 - ...dann Vokabeln zu Sätzen kombinieren...

Programmiersprache

- Sprache für Computer
 - Führt aus
 - [Manchmal: schreibt]
- Sprache für Menschen
 - Schreibt und liest
 - [Früher: führt aus]



Heute: Maschine
Früher: Beruf

Charles [Peirce] was appointed to his first official position in the Coast Guard Survey [...] in July 1861, as an **assistant computer** at \$ 35 per month.

(J. Brent, Charles Sanders Peirce: A Life. 1993. p 61)

Programmiersprache in EProg

- Die meisten Konzepte sind unabhängig von der Sprache!
- Ohne Programmiersprache können wir aber nicht programmieren.
- Wir verwenden **Java™**
 - Industrial strength
 - viele Konzepte
 - Nicht alle in EProg verwendet
 - Mehr Themen/Konzepte in weiteren Vorlesungen
 - LTS-Version: Java 21
 - keine zu Versionen-spezifischen Konzepte: fast alles auch bereits in Java 11