

InCode – Einfach Programmieren



Teilnehmer:	Ben Siebert (15), Lukas Birke (15)
Erarbeitungsort:	Hattingen
Projektbetreuerin:	Frau Dr. Sabine Schmidtseifer-Sürig
Thema des Projekts:	Einfaches Programmieren
Fachgebiet:	Mathematik/Informatik
Wettbewerbssparte:	Schüler experimentieren
Bundesland:	Nordrhein-Westfalen
Wettbewerbsjahr:	2022

InCode – Einfach Programmieren

Ein Projekt von Ben Siebert und Lukas Birke

1. Kurzfassung

InCode ist eine Programmiersprache, welche wir entwickelt haben, damit Anfänger ein leichter Einstieg in die grundlegenden Konzepte der Programmierung gelingt. In InCode entwickelt der Benutzer mit Hilfe von deutschen Sätzen eigene Websites. Bei der Entwicklung haben wir besonderen Fokus auf die Einfachheit der Sprache gelegt, sodass sie nicht nur leicht zu verstehen ist, sondern auch einfach geschrieben werden kann. Eine Demo unseres Projektes kann unter <https://incodelang.de> eingesehen werden.



2. Inhaltsverzeichnis

Seite

1. Kurzfassung	1
2. Inhaltsverzeichnis	2
3. Einleitung	3
4. Vorgehensweise und Methoden	3
a. Funktionsweise der Sprache	3
b. Beispiele der Sprache (InCode zu JavaScript)	4
c. Der Editor	5
i. Projekte	6
ii. Projekte exportieren	7
iii. Projekte/Vorlagen veröffentlichen	7
iv. Playground	8
v. Personalisierung	8
vi. Zusammen entwickeln	9
vii. Mögliche Probleme	9
d. Command Line Interface	10
e. Implementierung von InCode	11
f. Dokumentation	11
5. Ergebnisse	12
6. Ergebnisdiskussion/Ausblick	12
7. Zusammenfassung	13
8. Quellenverzeichnis	13
9. Fortschritt seit dem Regionalwettbewerb	15

3. Einleitung

Die Motivation für unser Projekt bekamen wir, da wir im Freundeskreis feststellten, dass viele Programmier-Einsteiger nicht gut mit der Syntax von herkömmlichen Programmiersprachen zurechtkommen und schnell überfordert sind. Wir glauben, dass es gerade für Anfänger leichter ist, die grundlegenden Konzepte der Programmierung zu erlernen, wenn man dies in seiner Muttersprache machen kann. Deswegen haben wir InCode entwickelt. Mit InCode kann jeder in seiner Muttersprache (aktuell wird Deutsch und Englisch unterstützt) eigene Webseiten entwickeln und währenddessen alle wichtigen Konzepte der Programmierung erlernen.

Besonderen Fokus haben wir daraufgelegt, dass der Code, den der Benutzer schreibt, aus möglichst grammatikalisch korrekten Sätzen besteht, und so gut gelesen werden kann. Wir haben nun also in insgesamt über 10.000 Zeilen Code ein geschlossenes System entwickelt, welches die Sprache an sich, einen Editor für die Sprache, ein Kommandozeilen-Werkzeug und viele weitere Unterprojekte enthält. Während der Entwicklung von InCode haben wir außerdem viele nützliche Open-Source Bibliotheken für JavaScript/TypeScript entwickelt, die uns bei der Programmierung unterstützt haben. Eine Liste aller Unterprojekte von InCode ist im Quellenverzeichnis zu finden. All unsere Projekte sind in JavaScript und/oder TypeScript geschrieben und können so plattformunabhängig ausgeführt werden. InCode kann unter <https://incodelang.de> verwendet werden.

4. Vorgehensweise und Methoden

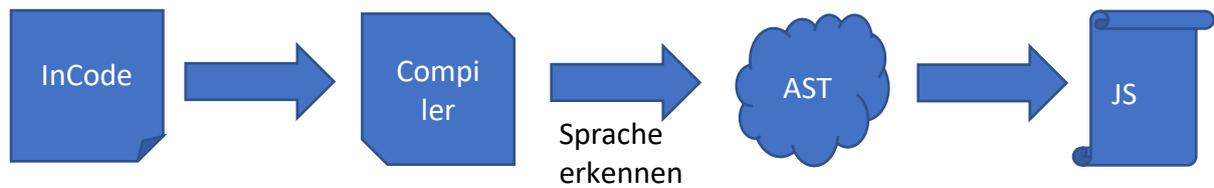
a. Funktionsweise der Sprache

InCode ist eine Programmiersprache, welche zu JavaScript kompiliert wird. Der generierte JavaScript Code läuft asynchron zum Rendering-Thread des Browsers, welches die Website darstellt. Dieses System hat den Vorteil, dass der Code ausgeführt wird, während die Website dargestellt wird (Rendering). Durch das unabhängige Ausführen dieser Aufgaben ist die programmierte Website sehr schnell.

Die erste Aufgabe des Compilers ist es, die Sprache (Deutsch/Englisch) automatisch zu erkennen. Hierzu wird der erste Befehl (Verb im Imperativ) des Skriptes mit einer Datenliste verglichen, die anzeigt, welcher Befehl zu welcher Sprache gehört. Anschließend lädt der Compiler die Daten für die entsprechende Sprache.

Als nächstes wird ein sogenannter Abstract Syntax Tree (AST, dt.: Abstrakter Syntax Baum) generiert. Dieser Abstract Syntax Tree kann nun vom Compiler verwendet werden, um den eigentlichen JavaScript Code zu generieren. Der Abstract Syntax Tree ist unabhängig von der verwendeten Sprache und kann so für jede aktuelle und zukünftige Sprache verwendet werden. Um ein Skript mit 10 Zeilen an Code zu kompilieren, benötigt der Compiler weniger als 0.008 Sekunden. Der TypeScript Compiler (tsc) braucht beispielsweise 2,34 Sekunden, um eine TypeScript-Datei mit der gleichen Größe zu kompilieren. Wenn der JavaScript Code fertig generiert wurde, gibt der Compiler den generierten Code zurück und beendet seine Arbeit.

Schematische Darstellung der Vorgehensweise bei InCode:



b. Beispiele der Sprache (InCode zu JavaScript)

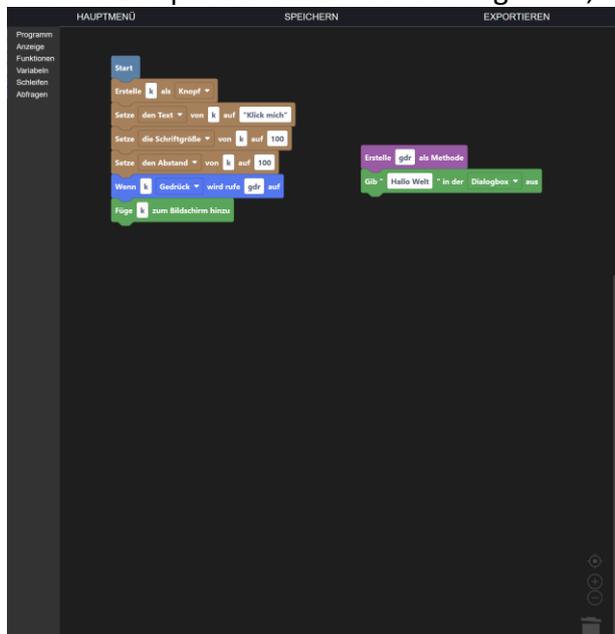
InCode	JavaScript
Gib „Hallo Welt“ in der Konsole aus	<code>console.log("Hallo Welt")</code>
Erstelle x	<code>let x</code>
Erstelle x als Knopf	<code>let x = document.createElement('button')</code>
Setze den Wert von x auf 2	<code>x = 2</code>
Setze den Text von x auf „Klick mich“	<code>x.innerText = "Klick mich"</code>
Wenn x gleich 2 ist	<code>if(x === 2)</code>
Warte 2 Sekunde	<code>await new Promise(done => setTimeout(() => done(), 2000))</code>
Warte 2000 Millisekunden	<code>await new Promise(done => setTimeout(() => done(), 2000))</code>
Setze die Farbe von x auf schwarz	<code>x.style.color = "black";</code>
Wenn x gedrückt wird rufe die Methode xGedrückt auf	<code>x.addEventListener('click', window.incode.xGedrückt);</code>
Erstelle die Methode xGedrückt Gib „Hallo Welt“ in der Konsole aus	<code>window.incode.xGedrückt = function () { console.log("Hallo Welt") }</code>
Erstelle x als Knopf Setze den Text von x auf „Klick mich“ Füge x zum Bildschirm hinzu	<code>let x = document.createElement('button'); x.innerText = "Klick mich" document.body.appendChild(x)</code>
Erstelle xPressed als Methode Gib „Hallo Welt“ in der Dialogbox aus Wenn x gedrückt wird rufe die Methode xPressed auf	<code>window.incode.xPressed = function () { alert("Hallo Welt"); } x.addEventListener('click', window.incode.xPressed);</code>

c. Der Editor

Für die Programmiersprache InCode haben wir außerdem einen eigenen Editor entwickelt, welcher es dem Benutzer erleichtern soll, eigene Programme mit InCode zu entwickeln. Damit der Editor ebenfalls international wie die Sprache, verwendet werden kann, haben wir den Editor in jede Sprache übersetzt, die von InCode unterstützt wird. Die Sprache des Editors kann nicht manuell verändert werden, sondern passt sich automatisch an die auf dem Computer verwendete Internet-Browser-Sprache an. Welche Sprache von InCode im Editor verwendet wird, ist jedoch unabhängig von der Editor-Sprache.

Im Editor kann entweder mittels Text oder Blöcken programmiert werden. Der Vorteil an der blockartigen Programmierung ist, dass der Benutzer keine Rechtschreibfehler machen kann, die dazu führen, dass die Website nicht mehr funktioniert.

Hier ein Beispiel für ein einfaches Programm, das mit dem Block-Editor entwickelt wurde:



The screenshot shows the InCode block editor interface. On the left, there is a sidebar with menu items: Programm, Anzeige, Funktionen, Variablen, Schichten, and Abfragen. The main workspace is divided into three sections: HAUPTMENÜ, SPEICHERN, and EXPORTIEREN. The visual programming flowchart consists of the following blocks: a 'Start' block, an 'Erstelle k als Knopf' block, a 'Setze den Text von k auf "Klick mich"' block, a 'Setze die Schriftgröße von k auf 100' block, a 'Setze den Abstand von k auf 100' block, a 'Wenn k gedrückt wird rufe gdr auf' block, an 'Erstelle gdr als Methode' block, a 'Gib "hallo Welt" in der Dialogbox aus' block, and a 'füge k zum Bildschirm hinzu' block. To the right of the editor, a preview window shows a button with the text 'Klick mich'.

Copyright © 2022 The InCode Develop

Und so sieht das gleiche Projekt als Text-Version aus:



The screenshot shows the InCode text editor interface. The main workspace is divided into three sections: HAUPTMENÜ, SPEICHERN, and EXPORTIEREN. The code is displayed as a list of 8 lines:

- 1 **Erstelle** k als knopf
- 2 **setze** den text von k auf "Klick mich"
- 3 **Setze** die Schriftgröße von k auf 100
- 4 **setze** den abstand von k auf 100
- 5 **erstelle** gdr als Methode
- 6 **Gib** "hallo Welt" in der Dialogbox aus
- 7 **wenn** k gedrückt wird rufe gdr auf
- 8 **füge** k zum bildschirm hinzu

To the right of the editor, a preview window shows a button with the text 'klick mich'.

i. Projekte

Mit der Hilfe des InCode Editors kann an mehreren Projekten gleichzeitig gearbeitet werden. Die Projekte sind mit dem Account des Benutzers verbunden, sodass die Projekte nicht nur auf einem Computer lokal verwendet werden können, sondern sich der Benutzer lediglich anmelden muss und direkt auf alle Projekte, die er erstellt hat, zugreifen kann.

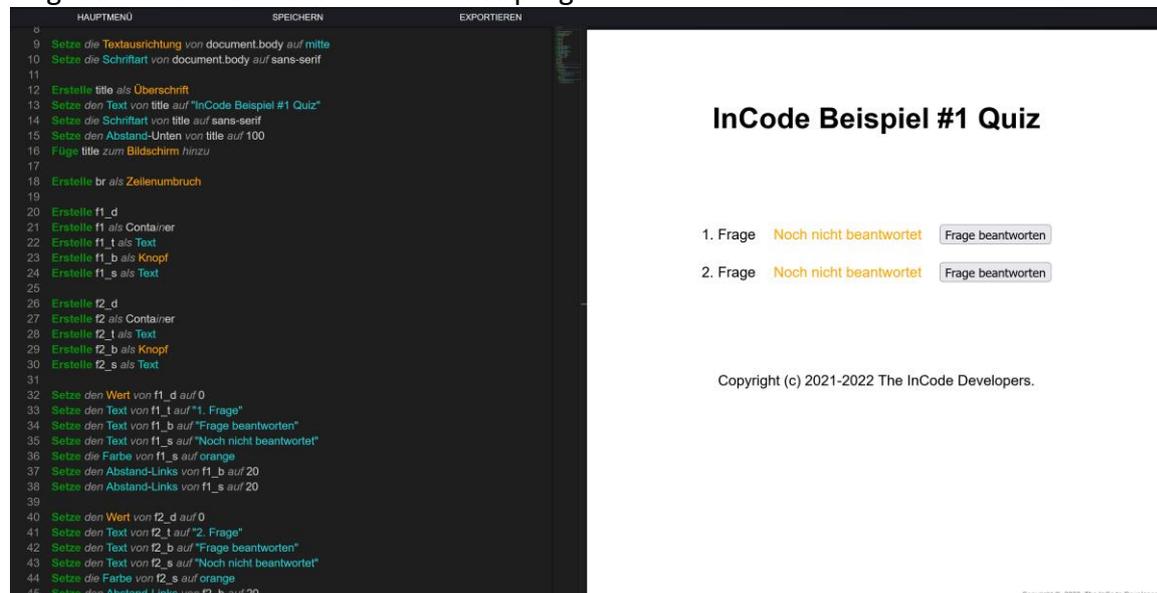


Jedes Projekt hat einen eigenen Namen, um wiedererkannt zu werden. Jeder Name kann nur einmal verwendet werden, er kann aber nach der Erstellung des Projektes zu jedem Zeitpunkt geändert werden. Projekte können ebenfalls als Vorlagen veröffentlicht werden, sodass andere Benutzer ein neues Projekt auf diesem aufbauen können.

Diese Vorlagen können nach der Veröffentlichung vom Autor verändert und neu hochgeladen werden. Die Änderungen werden allerdings nicht auf andere Projekte, die auf diesem aufbauen angewandt, sondern nur, wenn neue Projekte mit dieser Vorlage erstellt werden.

Außerdem kann man einen Link für sein Projekt erstellen, unter dem jeder dieses Projekt ausprobieren kann. Diese Links können weder entfernt, noch bearbeitet werden. Mit dem InCode Editor können viele verschiedene Projekte entwickelt werden.

Ein beispielhaftes Quiz-Projekt ist unter <https://ex.incode-lang.de/quiz/> zu finden. Der Code dieses Beispielprojektes kann unter <https://ex.incode-lang.de/quiz/code.ic> eingesehen werden. Auf der linken Seite der untenstehenden Abbildung ist ein Ausschnitt des Codes dieses Projektes zu sehen und auf der rechten Seite ist zu sehen, wie die programmierte Website aussehen wird. Die rechte „preview“ Seite aktualisiert sich alle 3 Sekunden und zeigt so immer den neusten Stand des programmierten Codes an.



ii. Projekte exportieren

InCode Projekte können mit Hilfe des Editors zu einer Website exportiert werden. Um den Editor möglichst einsteigerfreundlich zu halten, muss hierfür lediglich ein Knopf gedrückt werden. Anschließend öffnet sich ein Datei-Download, welcher eine ZIP-Datei enthält. Diese ZIP-Datei wird im Browser des Benutzers generiert und enthält eine HTML-Datei, eine JavaScript-Datei, eine InCode-Datei und eine Text-Datei.



Die HTML-Datei ist die Website an sich, welche doppelt geklickt werden kann, um die Website zu öffnen. In der JavaScript-Datei ist die Logik der Website enthalten, welche die Elemente auf die Website bringt. Prinzipiell enthält die JavaScript-Datei nur den Programm-Code, welcher zuvor kompiliert wurde. In der Text-Datei befinden sich einige Informationen zu den anderen Dateien. Die InCode-Datei beinhaltet den ursprünglichen Programm-Code.

Das exportierte Projekt ist am einfachsten auf einem Computer zu verwenden. Auf Endgeräten wie Tablets oder Handys ist dies zwar auch möglich, jedoch nicht sehr benutzerfreundlich. Das exportierte Projekt kann ohne Probleme direkt auf einen Server hochgeladen werden, um so eine Website zu veröffentlichen.

iii. Projekte/Vorlagen veröffentlichen

InCode Projekte können als Vorlage veröffentlicht werden, sodass andere Benutzer ein neues Projekt mit dem ursprünglichen als Grundlage erstellen können. Nachdem eine Vorlage veröffentlicht wurde, kann diese nur noch vom ursprünglichen Autor bearbeitet und

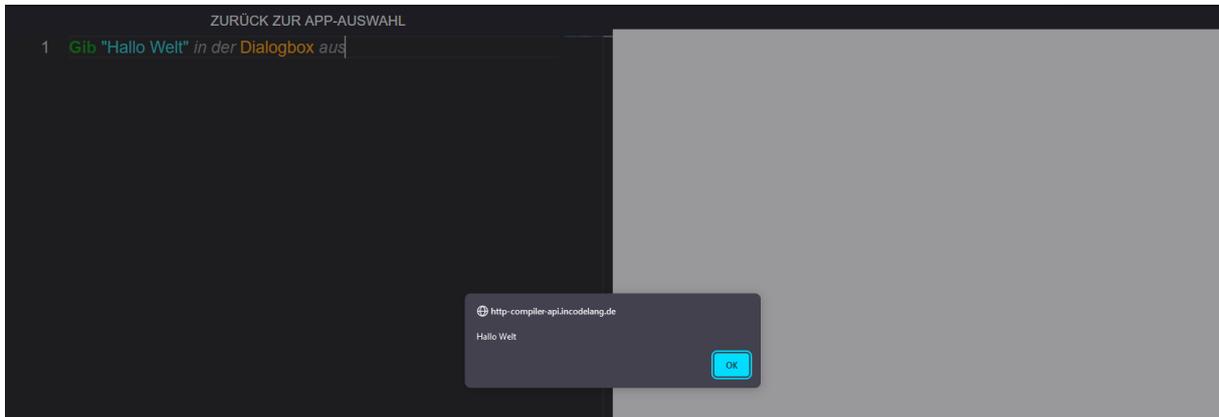


wieder gelöscht werden. Sollte eine neue Version einer Vorlage veröffentlicht worden sein, werden die Änderungen nicht in allen Projekten übernommen, die auf dieser Vorlage aufbauen, sondern nur bei neuen Projekten angewandt. Projekte können aber ebenfalls veröffentlicht werden. Hierbei wird zunächst ein Link erstellt, bei dessen Aufruf eine Website mit dem kompilierten Programm-Code gestartet wird. Dieser Link

wird als nächstes mit dem integrierten URL-Kürzer gekürzt, sodass er besser weiter verschickt werden kann. Diese Links können weder bearbeitet noch gelöscht werden.

iv. Playground

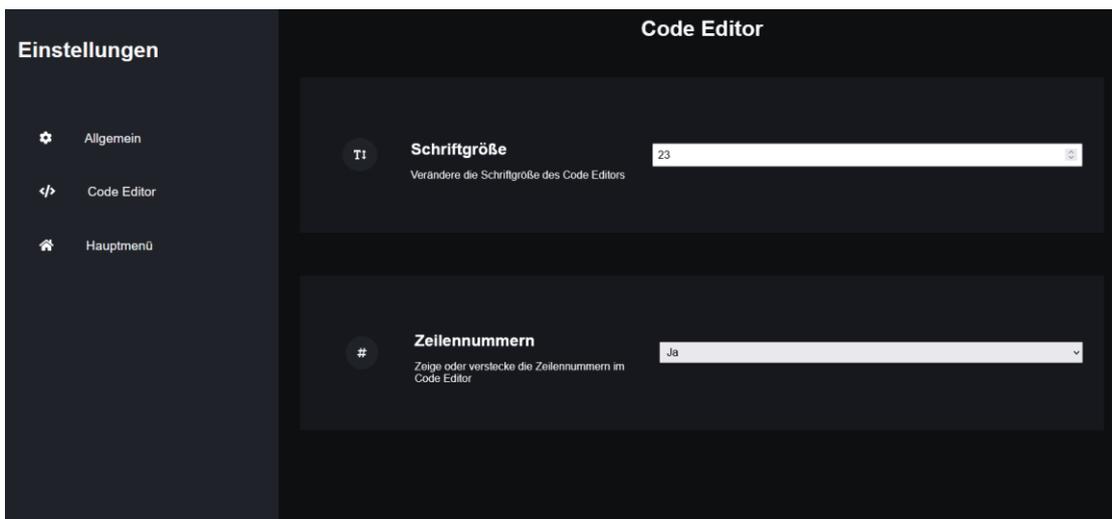
Der InCode Editor verfügt ebenfalls über einen Playground, d.h. eine Testumgebung ohne Speichermöglichkeit. In diesem Playground kann der Benutzer InCode testen, ohne einen Account zu erstellen oder sich anmelden zu müssen. Dieser Playground eignet sich sehr gut, um schnell einen Befehl auszuprobieren, jedoch ist er absichtlich nicht für Projekte geeignet, da nichts gespeichert wird und alles nach einem erneuten Laden der Seite verloren ist.



Beim Playground lassen sich außerdem keine Projekte exportieren und auch keine Vorlagen veröffentlichen. Zudem bietet der Playground keine Möglichkeit mit Blöcken zu programmieren. In der Zukunft wäre es aber durchaus denkbar, dass wir den Block-Editor auch im Playground integrieren.

v. Personalisierung

Benutzer können den InCode Editor in vielerlei Hinsicht personalisieren und auf ihre individuellen Vorstellungen anpassen. Aktuell (Abgabezeitpunkt der schriftlichen Arbeit)



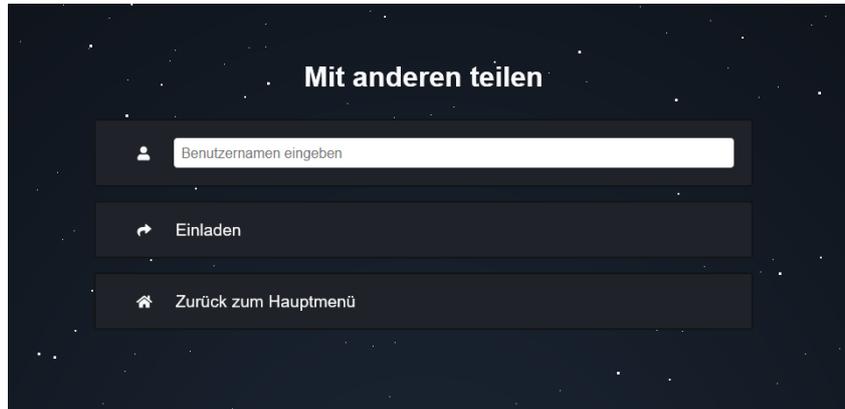
lassen sich lediglich die Schriftgröße, sowie die Zeilennummern im Code-Editor anpassen. Zukünftig wird der Benutzer noch das Erscheinungsbild (Hell/Dunkel/Hoher-Kontrast) verändern können.

vi. Zusammen entwickeln

Der Editor bietet ebenfalls die Möglichkeit, mit mehreren Benutzern zusammen an einem Projekt zu programmieren. Hierfür muss der Ersteller des Projektes einen anderen Benutzer einladen und ihm somit kompletten Zugriff auf das Projekt geben. Nach dem die Einladung verschickt wurde, sieht der eingeladene Benutzer diese in den Account Einstellungen. Hier kann er sie akzeptieren und dem Projekt beitreten. Momentan gibt es bei der

Zusammenarbeit an einem Projekt noch das Problem, dass es keine Live-Synchronisierung gibt, sodass beim Speichern der Fortschritt anderer Benutzer überschrieben werden kann. Aktuell arbeiten wir an einer Lösung, bei der dieses Problem mittels einer Live-Synchronisierung

gelöst wird. Hierbei werden dann, sobald ein Benutzer etwas am Projekt verändert, die Änderungen in Echtzeit an jeden anderen Benutzer weitergeleitet. Da diese Lösung des Problems sehr anspruchsvoll und aufwändig ist, wird die endgültige Fertigstellung etwas Zeit in Anspruch nehmen.



vii. Mögliche Probleme

Bei der Entwicklung einer Website mit InCode kann es zu zwei Problemen kommen.

Das erste Problem ist, dass der gesamte Website Code in einer einzigen InCode-Datei geschrieben werden muss. So kann es bei größeren Projekten schnell unübersichtlich werden. In Zukunft ist es aber geplant, den Website Code in mehrere InCode-Dateien aufzuteilen, sodass der Code der Website übersichtlicher gestaltet werden kann.

Das zweite Problem ist, dass die generierte Website nur in modernen Browsern, wie Firefox, Chrome, Opera oder Edge, dargestellt werden kann, da InCode den ECMA-Script Standard ES8/ES2017 nutzt, welcher von Browsern wie dem Internet Explorer nicht unterstützt wird. Welche Browser diesen Standard genau unterstützen kann unter <https://kangax.github.io/compat-table/es2016plus/> (ab „2017 features“) eingesehen werden.

d. Command Line Interface

Für InCode haben wir ebenfalls eine Kommandozeilen Anwendung (CLI) entwickelt, welche fortgeschrittene Benutzer verwenden können, um InCode außerhalb des Editors zu kompilieren. Das CLI kann im Vergleich zum Editor sogar die AST Dateien speichern und kompilieren. Wichtig ist, dass dieses Unterprojekt nicht für unerfahrene Benutzer empfohlen ist, da es nur eine Konsolenanwendung ist, die über keine grafische Benutzer Oberfläche (GUI) verfügt. Der Benutzer muss mittels Befehlen in der Konsole das Programm steuern.

```
PS C:\Users\bensi> incode --help
Usage: incode [options]

Options:
  -V, --version          output the version number
  -i, --input <file>    Specify the InCode file you want to compile.
  -a, --save-ast         Save the Abstract-Syntax-Tree File.
  -o, --output <file>  Set the output file.
  -s, --std-out         Print the compiled code to the Standard Output. (default: false)
  -f, --force           Force overwrite the output file if it exists. (default: false)
  -h, --help           display help for command
PS C:\Users\bensi> |
```

Hier eine Auflistung aller Optionen, die der Benutzer verwenden kann, um das Programm zu steuern:

Option	Beschreibung	Notwendig/Optional
-i, --input <Datei>	Setzt die Quelldatei, die kompiliert werden soll (Es wird automatisch erkannt, ob die eingegebene Datei Quellcode oder einen AST enthält)	<i>Notwendig</i>
-o, --output <Datei>	Setzt die Ausgabedatei, in der der kompilierte Code gespeichert werden soll	<i>Notwendig</i>
-f, --force	Überschreibt die Ausgabedatei, wenn diese bereits existiert	<i>Optional</i>
-a, --save-ast	Speichert die AST Dateien (Abstract Syntax Tree)	<i>Optional</i>
-s, --std-out	Gibt den kompilierten Programm-Code in der Standard-Ausgabe aus (stdout)	<i>Optional</i>

e. Implementierung von InCode

InCode ist in viele einzelne Module aufgeteilt, die unabhängig voneinander verwendet werden können. So ist es beispielsweise möglich, den Compiler in seinem eigenen Projekt zu verwenden. Ebenfalls macht es der Aufbau möglich, kleinere Module, wie den URL-Kürzer oder das Account-System im eigenen Projekt zu verwenden.

InCode ist in viele NPM-Packages aufgebaut und kann so mittels des „Node Package Manager“ (NPM) einfach in jedes NodeJS Paket importiert werden. Dafür haben wir eine eigene NPM Organisation angelegt, welche „@incodelang“ benannt wurde. Um also eines unserer Projekte zu installieren, muss man zunächst den „Node Package Manager“ installieren und kann anschließend mittels des Befehls „`npm install @incodelang/<paket_name>`“ das entsprechende Paket installieren. So lautet der Befehl, welcher den Compiler installiert beispielsweise „`npm install @incodelang/compiler`“. Damit man die Sprache auch in einem Projekt verwenden kann, welches nicht den „Node Package Manager“ oder „yarn“ verwendet, haben wir eine REST-API entwickelt, welche dazu genutzt werden kann, InCode-Code zu kompilieren. Diese REST-API kann unter <https://http-compiler-api.incodelang.de> aufgerufen werden. Eine Dokumentation zu der Bedienung dieser API ist unter <https://GitHub.com/InCodeDevs/HTTP> zu finden.

f. Dokumentation

InCode verfügt über eine detaillierte Dokumentation, in der sowohl alle Befehle als auch Eigenschaften mit Beispielen zu finden sind. Hiermit soll der Einstieg in InCode weiter vereinfacht werden: Neben der Funktion als Nachschlagewerk wird die Dokumentation auch geeignet sein, neue Befehle kennenzulernen und deren Aufbau und Anwendungsmöglichkeiten zu verstehen.

Diese Dokumentation ist momentan noch nicht vollkommen fertiggestellt, wird aber stetig erweitert und an die neusten Funktionen von InCode angepasst.

Befehl: Rufe

Kurzfassung
Der Rufe Befehl kann dazu verwendet werden, eine Methode aufzurufen.

Syntax

Syntax	Bedeutung
Rufe die Methode <Name> auf	Ruft die Methode mit dem Namen <Name> auf

Beispiele

Ruft die Methode m1 auf

```
Rufe die Methode m1 auf
```

Ruft die Methode meine_funktion auf

```
Rufe die Methode meine_funktion auf
```

5. Ergebnisse

Während der Entwicklung von InCode haben wir viele Ergebnisse erzielt. Hier eine Auflistung der wichtigsten Ergebnisse:

- Es ist nun möglich mit der Hilfe unserer eigenen Programmiersprache, unter der Verwendung der Muttersprache des Benutzers, eigene Webseiten zu entwickeln. Unsere Programmiersprache ist sehr leicht zu lesen und zu schreiben, sodass ein sehr leichter Einstieg möglich ist.
- Mit dem Editor haben wir eine Möglichkeit geschaffen, mehrere Projekte zu verwalten. Diese Projekte können als Vorlage veröffentlicht werden, sodass andere Benutzer ein neues Projekt entwickeln können, aber das vorherige als Basis verwenden können. Außerdem ist es möglich, ganze Projekte zu veröffentlichen, sodass die programmierten Webseiten mit wenig Zeitaufwand mit Freunden und der Familie geteilt werden können.
- Der Editor ermöglicht außerdem, dass mehrere Benutzer am gleichen Projekt arbeiten können. Wenn ein Benutzer etwas an diesem geteilten Projekt ändert, aktualisiert sich das Projekt bei allen anderen Benutzern, die Zugriff auf dieses Projekt haben.
- Wir haben sowohl InCode als auch den InCode Editor in Deutsch als auch in Englisch übersetzt, sodass InCode international verwendet werden kann.
- Mit der InCode Dokumentation ist es neuen Benutzer möglich, die Syntax von InCode zu erlernen. Außerdem kann diese Dokumentation von erfahreneren Benutzern als Nachschlagwerk verwendet werden.
- Der Editor verwendet ein modernes Account System, sodass die Projekte nicht auf einem Computer gespeichert werden, sondern an den Account des Benutzers geknüpft sind. So kann auch von einem anderen Endgerät an einem Projekt weitergearbeitet werden.
- Durch den integrierten Block-Editor ist es dem Benutzer möglich InCode auch ohne Text schreiben zu müssen zu verwenden.

6. Ergebnisdiskussion/Ausblick

In der Zukunft werden wir die Möglichkeiten von InCode erweitern. So werden wir beispielsweise die Option hinzufügen, InCode Projekte in mehrere Dateien aufzuteilen. Außerdem werden wir den Block-Editor in den Playground integrieren, sodass man im Playground nicht mehr nur mit Text programmieren kann. Des Weiteren werden wir eine Live-Synchronisierung für die Zusammenarbeit entwickeln.

Wir planen zudem noch eine Desktop/Handy/Tablet fähige Version des Editors zu entwickeln, welche auch ohne Internetzugriff funktionieren soll.

Auch weitere Personalisierungsmöglichkeiten, die in den Einstellungen angepasst werden können, sollen noch ergänzt werden.

Parallel ergänzen und vervollständigen die Dokumentation weiter.

Aktuell gibt es in der InCode Sprache noch einige wenige Fehler, die wir beheben müssen, wie zum Beispiel, dass die Einrückungen (TABs) nicht richtig erkannt werden.

7. Zusammenfassung

Zusammenfassend ist InCode ein Projekt, welches gerade Programmierneulingen dabei helfen kann, die grundlegenden Aspekte der Programmierung zu lernen. Das Wissen über Methoden, Schleifen, Abfragen und Variablen ist essenziell für die Programmierung in fast jeder Programmiersprache, sodass die gewonnenen Erkenntnisse auf andere Programmiersprachen, wie zum Beispiel Python oder JavaScript angewendet werden können. Der Editor hilft dem Benutzer in vielerlei Hinsicht, einfacher und schneller InCode zu programmieren und verschiedene InCode Projekte zu verwalten. Der Benutzer kann von jedem Gerät, das über eine Internetverbindung verfügt, auf seine Projekte zugreifen und so auch von unterwegs an ihnen weiterarbeiten.

Da der gesamte InCode Quellcode Open-Source auf GitHub unter <https://github.com/InCodeDevs> zu finden ist, kann InCode auch als Basis für neue Programmiersprachen verwendet werden. Der Editor ist ebenfalls so programmiert, dass er problemlos auf andere Programmiersprachen angepasst werden kann. Während der Entwicklung von InCode haben wir viele Open-Source Bibliotheken entwickelt, welche heute in anderen Projekten genutzt werden können. So ist es zum Beispiel mit unserer Account Bibliothek, welche wir ursprünglich für den Editor entwickelt haben, möglich, ein sicheres Account-System einfach in ein anderes Projekt zu implementieren. Alle Open-Source Projekte von uns sind unter der *GNU General Public License 3.0 (GPLv3)* veröffentlicht, sodass sie auch im kommerziellen Umfeld verwendet werden können. Die genauen Spezifikationen dieser Open-Source Lizenz können unter <https://choosealicense.com/licenses/gpl-3.0> eingesehen werden.

8. Quellenverzeichnis

Bildquellen:

- Alle Grafiken wurden von uns erstellt.

Nützliche Links:

- Website des Projektes: <https://incodelang.de>
- Dokumentation der Sprache: <https://incodelang.de/docs>
- InCode Editor: <https://incodelang.de/editor>

Code-Repositories:

- InCode: <https://GitHub.com/InCodeDevs/Compiler>
- InCode Editor: <https://GitHub.com/InCodeDevs/Editor>
- InCode CLI: <https://GitHub.com/InCodeDevs/CLI>
- URL-Shorter: <https://GitHub.com/InCodeDevs/URLShorter>
- Accounts: <https://GitHub.com/InCodeDevs/Accounts>
- Accounts Web Client: <https://GitHub.com/InCodeDevs/Accounts-Client>
- Andere: <https://GitHub.com/InCodeDevs>

Verwendete Open-Source Bibliotheken (Editor):

Name	Version
@types/react	17.0.37
@types/react-dom	17.0.11
@typescript-eslint/eslint-plugin	5.5.0
@typescript-eslint/parser	5.5.0
concurrently	6.4.0
copy-webpack-plugin	10.0.0
css-loader	6.5.1
eslint	8.3.0
file-loader	6.2.0
html-webpack-plugin	5.5.0
husky	7.0.4
node-sass	6.0.1
prettier	2.5.0
sass-loader	12.3.0
style-loader	3.3.1
ts-loader	9.2.6
typescript	4.5.2
url-loader	4.1.1
webpack	5.64.4
webpack-cli	4.9.1
@fortawesome/fontawesome-svg-core	1.2.36
@fortawesome/free-brands-svg-icons	5.15.4
@fortawesome/free-regular-svg-icons	5.15.4
@fortawesome/free-solid-svg-icons	5.15.4
@fortawesome/react-fontawesome	0.1.16
@monaco-editor/react	4.3.1
blockly	6.20210701.0
body-parser	1.19.0
chalk	4.1.2
cors	2.8.5
dotenv	10.0.0
express	4.17.1
jsdom	19.0.0
jszip	3.7.1
monaco-editor	0.30.1
react	17.0.2
react-dom	17.0.2
serve-index	1.9.1
web-push	3.4.5

9. Fortschritt seit dem Regionalwettbewerb

Seit dem Regionalwettbewerb am 16. Februar 2022 haben wir InCode um viele Funktionen erweitert.

So ist jetzt jede Zeile des JavaScript-Programmcodes mit einem Kommentar versehen. Dieser Kommentar erklärt die nachfolgende JavaScript-Code Zeile und hilft so dem Benutzer, den finalen Code besser zu verstehen. Dieses Vorgehen erleichtert den Umstieg von InCode zu JavaScript, da der Benutzer so verstehen kann, welche JavaScript-Code Zeile was macht.

Außerdem haben wir Synonyme hinzugefügt, sodass statt „Erstelle“ beispielsweise auch „Definiere“ verwendet werden kann. Dies macht die Syntax flexibler, was es vor allem Anfängern erleichtert, InCode zu lernen, da sie nicht an eine feste Syntax gebunden sind. *Hinweis: Der Satzbau ist aktuell nicht veränderlich, jedoch arbeiten wir an dieser Funktion.*

Des Weiteren haben wir InCode um den „Entferne“-Befehl erweitert, welcher es ermöglicht Elemente, die zuvor zum Bildschirm hinzugefügt worden sind, vom Bildschirm zu entfernen.

Die größte Änderung, die wir vorgenommen haben, ist, dass der Benutzer nun seine InCode-Programme als Desktop-App exportieren kann. Exportiert wird das Programm mit Hilfe des Electron-Frameworks (<https://electronjs.org>), welches ebenfalls eine installierbare Version der App (Windows-Installer) bereitstellt. Diese Funktion kann jetzt schon unter <https://incodelang.de> ausprobiert werden, jedoch ist sie zurzeit noch in der Test-Phase. Aktuell funktionieren die exportierten Desktop-Apps nur unter Computern, die das Windows Betriebssystem verwenden. Wir arbeiten daran, dass Desktop-Apps auch unter MacOS und Linux verwendet werden können.

Die in Kapitel 6 (Ergebnisdiskussion/Ausblick) erwähnte Live-Synchronisierung haben wir ebenfalls hinzugefügt. So ist es Benutzern nun möglich, gleichzeitig an einem Projekt zu arbeiten und in Echtzeit die Änderungen der anderen Benutzer zu sehen. Diese Funktionalität erleichtert die Zusammenarbeit insofern, dass nun mehrere Benutzer gleichzeitig ein Projekt editieren können, ohne die Änderungen der anderen Benutzer zu überschreiben oder ggf. zu löschen.

Außerdem haben wir noch einige kleine Änderungen vorgenommen. So haben wir ein Admin-Panel hinzugefügt, welche es Systemadministratoren erleichtert, die Website zu verwalten. Zudem haben wir das Design an einigen Stellen (wie z.B. Popups) überarbeitet.