

# Entwicklungsumgebungen für Informatik-Anfänger

Sabine Terwelp, Markus Dahm

Fachbereich Medien, FH Düsseldorf

## **Zusammenfassung**

Um Anfängern, insbesondere Kindern und Schülern, das Erlernen von „Programmieren“ zu erleichtern, wird eine Reihe von speziellen Entwicklungsumgebungen (IDE) angeboten. Sie bilden die Brücke in die neue und abstrakte Welt des Programmierens. Der Lernerfolg hängt damit nicht unwesentlich davon ab, wie verständlich die IDE selber ist, wie wenig Hindernisse sie dem Erlernen in den Weg legt und wie gut wesentliche Programmier-Konzepte, insbesondere die der Objektorientierung, verdeutlicht werden. In einer Marktstudie wurden daher IDEs auf Ihre Lerntauglichkeit untersucht (Java Hamster, Greenfoot, Storytelling Alice, Scratch und Kara). Die Ergebnisse finden sich in diesem Beitrag. Aufbauend u.a. auf diesen Ergebnissen wird in der weiteren Forschung eine eigene IDE entwickelt.

## 1 Motivation

Programmier-Anfänger müssen zu Anfang auf vielerlei neuen Ebenen kognitive Leistungen erbringen: Die eher abstrakten Grundkonzepte des Programmierens, die Syntax der konkreten Programmiersprache und die Nutzung der Entwicklungsumgebung (IDE) müssen verstanden sein, bevor man sich dem Verstehen der gestellten Aufgabe sowie endlich dem eigentlichen Lernziel, nämlich der Lösung der Aufgabe in diesem Kontext widmen kann.

Um diese Vielzahl an neuen und ungewohnten Feldern nicht zu allzu großen Hindernissen anwachsen zu lassen, gibt es einige Entwicklungsumgebungen, die speziell für Programmieranfänger gedacht und gemacht sind. Ihnen ist gemeinsam, dass Sie nicht alle Möglichkeiten einer Programmiersprache mit allen verfügbaren Libraries und Features in einer professionellen Umgebung bieten, sondern nur einen kleinen Ausschnitt bieten. Fast immer wird an die Erfahrungen und die Lebenswelt der Anfänger angeknüpft, indem eine einfache grafische „Welt“ dargestellt wird, in der wenige Objekte vorhanden sind und durch eigene Programmierung gesteuert werden können. Die IDEs unterscheiden sich u.a. durch die Art und Weise, wie Code definiert wird: in Anlehnung an bzw. Nutzung einer bestehenden Programmier- oder Scriptsprache (z.B. Java) mittels Text oder grafisch mit drag and drop. Außerdem gibt es prozedurale, Objekt- oder Automaten-orientierte IDEs.

## 2 Die untersuchten IDEs

Es wurden die IDEs Java Hamster, Greenfoot, Storytelling Alice, Scratch und Kara untersucht um ihre Tauglichkeit für die Lernunterstützung zu ermitteln (Terwelp 2011). Gemeinsam ist, dass sie für Programmier-Anfänger konzipiert wurden und die Programmierung von Objekten in einer grafischen 2D- bzw. 3D-Welt visualisiert wird.

### 2.1 Java Hamster und Greenfoot

Die IDE „Java Hamster“ (Abb. 1) wurde für junge Programmieranfänger an der Universität Oldenburg entwickelt (JH 1998). Sie bietet ein Editor- und ein 2D-Simulations-Fenster, auf welchem ein „Hamster“ mit Java-Code gesteuert werden kann.

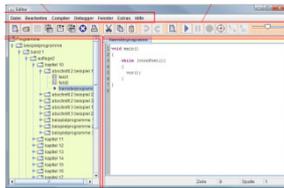


Abb. 1 – Java Hamster: Editor- und Simulations-Fenster

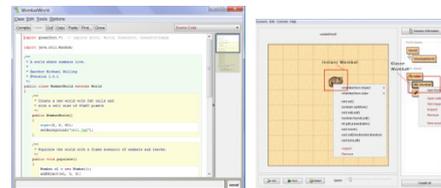


Abb. 2 – Greenfoot: Editor- und Welt-Fenster

Greenfoot (GF 2010) basiert auf der Java-IDE BlueJ und wurde um eine 2D-Welt erweitert, die der Simulation des Java-Hamsters sehr ähnlich ist (Abb. 2). Es werden Objekte in die Welt eingesetzt, die dann alleine oder miteinander agieren können. Die vorhandenen Methoden können angepasst oder ergänzt werden. Auch ganz neue Methoden können eingefügt und interaktiv aufgerufen werden. Java- bzw. Programmierkenntnisse sind nötig.

In beiden IDEs wird in Java-Syntax programmiert. Der Editor bietet jeweils Unterstützung mit einem Klassenbrowser und Syntax-Highlighting.

### 2.2 Scratch und Storytelling Alice

Sowohl Scratch (SC 2011) als auch Storytelling Alice (SA 2007) sind stark auf narrative Aspekte ausgerichtet. Ziel ist, „interaktive Geschichten, Animationen, Spiele, Musik-, und

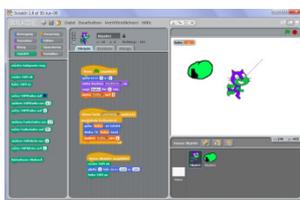


Abb. 3 – Scratch: IDE



Abb 4 - Storytelling Alice: IDE

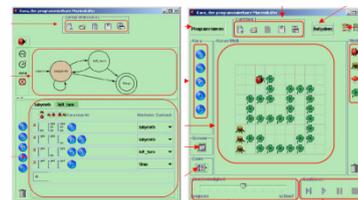


Abb. 5 – Kara Editor und Simulation

Kunstwerke zu erstellen“ (SC 2011). Es wird nicht durch Schreiben von Text programmiert, sondern durch blockweises Zusammenfügen von Bausteinen mittels drag and drop. Als Text geschriebener (z.B. Java-) Code kann nicht eingebunden werden. Scratch ist 2D-orientiert, Storytelling Alice stellt eine einfache 3D-Authoring-IDE zur Verfügung.

## 2.3 Kara

In Kara (KA 2006) werden Zustände und Übergangsbedingungen für endliche Automaten definiert (Abb. 5). Code oder Kommentar als Text zu schreiben ist nicht möglich. Dafür existieren Varianten wie javakara oder pythonkara, die Java-Hamster oder Greenfoot ähneln.

## 3 Bewertung und Ergebnisse

Diese IDEs wurde in folgenden Kategorien untersucht, die sich an den klassischen Themen und Richtlinien des Informatik-Unterrichts für Anfänger orientieren (z.B. NRW 2009, Hubwieser 2007, Frey 2004) und die Attraktivität und Gebrauchstauglichkeit einbeziehen.

1. Erläuterung von und Umgang mit Datentypen und Datenstrukturen
2. Erläuterung von und Umgang mit Kontrollstrukturen
3. Erläuterung von und Umgang mit Konzepten der Objektorientierten Programmierung
4. Attraktivität: Eigene Kreativität, mehrere Schwierigkeitsgrade, Unterrichtstauglichkeit
5. Gebrauchstauglichkeit: Installation, Erlernbarkeit, Übersichtlichkeit, Fehlerfindung

Für die Kategorien 1 bis 3 wurde überprüft, wie Programmieraufgaben ohne Vorkenntnisse bearbeitet werden. In der Tabelle unten sind gerundete Summen von mehreren gerundeten Einzelwertungen angegeben. Details zu allen Wertungen finden sich in (Terwelp 2011).

	<i>max. Pkt.</i>	Java Hamster	Greenfoot	Storytelling Alice	Scratch	Kara
<i>Daten-Typen, Strukturen</i>	20	7	7	19	20	5
<i>Kontrollstrukturen</i>	20	11	9	17	19	12
<i>Objektorientierung</i>	20	20	20	20	14	14
<i>Attraktivität</i>	20	11	13	16	17	14
<i>Gebrauchstauglichkeit</i>	20	12	15	18	19	16
<i>Gesamtbewertung</i>	100	61	64	89	88	61

Sowohl Storytelling Alice als auch Scratch sind gut geeignet, vor allem jungen Anfängern einen interessanten Einstieg in die Welt des Programmierens zu geben. Sie sind attraktiv und fördern die persönliche Kreativität. Man kann schnell und einfach mit ihnen spielerisch Programmierung ausprobieren und sofort erfolgreich Ergebnisse sehen. Mit diesen IDEs können

Grundprinzipien der Programmierung von Programmieranfängern gut alleine erarbeitet werden. Für einen Umstieg auf eine Text-orientierte Programmierung, wird jedoch keine Unterstützung, z.B. durch Code-Export oder Anleitung, angeboten.

Mit Java Hamster und Greenfoot können alle getesteten Aufgaben erfüllt werden. Jedoch bieten sie für Anfänger ohne Java-Kenntnisse und begleitenden Lehrer einige Hürden und Grenzen. Zudem ist ihre doch sehr simple Grafikausgabe zwar funktional, aber nicht (mehr) sehr attraktiv. Im speziellen Ansatz von Kara, endliche Automaten zu programmieren, gibt es für Anfänger zwar Hilfestellungen, die Kreativität ist aber eher eingeschränkt. Diese drei IDEs eignen sich gut als unterrichtsbegleitende Werkzeuge. Außerdem bieten Sie einen einfachen Übergang zur üblichen Text-orientierten Programmierung. Für den unbegleiteten Einsatz sind diese drei Tools für Anfänger eher nicht so gut geeignet.

Die Lernunterstützung besteht bei allen betrachteten IDEs aus meist sehr ausführlichen Dokumentationen, teilweise sehr guten Tutorials und teilweise nur sehr knappen integrierten Hinweisen zu den grundlegenden Konzepten der Nutzung der IDE.

## 4 Weitere Forschung

Was erstaunlicherweise keine der getesteten IDEs bietet (und auch umfangreichere IDEs eher rudimentär bieten), ist Hilfe auf dem Weg vom Problem zum Programm – obwohl dieser Weg sowohl theoretisch (Hubwieser 2007) als auch formal (NRW 2009) ein zentrales Lernthema ist. Weder in den IDEs noch in deren zusätzlichen Materialien wird auch nur ansatzweise eine strukturierte Vorgehensweise unterstützt. Auf dieses Feld konzentriert sich unsere weitere Forschung mit dem Ziel der Entwicklung und Evaluation einer IDE.

### Literaturverzeichnis

Frey (2004). *Informatik 1 für Gymnasien – Objekte, Strukturen, Algorithmen*, Klett, Stuttgart

GF (2010). *Greenfoot* <http://www.greenfoot.org>

Hubwieser (2007). *Didaktik der Informatik*, Springer, Berlin

JH (2011) *Java Hamster*. <http://www.java-hamster-modell.de>

KA (2006) *Kara*. <http://www.swisseduc.ch/informatik/karatojava/kara>

NRW (2009) *Richtlinien Informatik NRW*. MSWWF NRW, SekI: Heft 3424, SekII: Heft 4725, 4725/1

SA (2007). *Storytelling Alice*. <http://alice.org/kelleher/storytelling>

SC(2011). *Scratch*. <http://scratch.mit.edu>

Terwelp (2011). *Tools für den Informatik-Unterricht in der Schule*. Bachelorarbeit, FH Düsseldorf

### Kontaktinformationen

FH Düsseldorf, FB Medien, WWW: [www.medien.fh-duesseldorf.de/dahm](http://www.medien.fh-duesseldorf.de/dahm),

E-Mail: [markus.dahm@fh-duesseldorf.de](mailto:markus.dahm@fh-duesseldorf.de) [sabineterwelp@googlemail.com](mailto:sabineterwelp@googlemail.com)