

B.II.11

Algorithmen – Visuelle Programmierung

Einführung in die Programmierung einfacher Apps mit App Inventor

Kolja Markwardt



© RAABE 2024

© insta_photos/iStock / Getty Images Plus

In diesem Unterrichtsmaterial lernen Ihre Schülerinnen und Schüler auf spielerische und selbstentdeckende Weise das Programmieren und Arbeiten in einer Entwicklungsumgebung kennen. Das Unterrichtsmaterial liefert Ihnen eine ausführliche Schritt-für-Schritt-Anleitung, die mit farbigen Grafiken zur besseren Verständlichkeit anschaulich und liebevoll ausgestattet ist. Erstellen Sie mit Ihrer Klasse eine breite Palette von Apps, die die Schülerinnen und Schüler im Anschluss auf Ihren eigenen Smartphones herunterladen und nutzen können und stellen Sie damit einen direkten Bezug zur Lebenswelt der Kinder her.

KOMPETENZPROFIL

Klassenstufe:	5
Dauer:	2–3 Unterrichtsstunden
Lernziele:	Die Lernenden sind in der Lage mithilfe von App Inventor® verschiedene einfache Apps zu programmieren.
Thematische Bereiche:	Programmieren in einer Entwicklungsumgebung mit vorgegebenen Bausteinen
Kompetenzbereiche:	Modellieren, Implementieren, Produzieren und Präsentieren



netzwerk
lernen

zur Vollversion

Inhaltsverzeichnis

1 App Inventor®	2
1.1 Start und Anmeldung	2
1.2 App ausführen	3
1.3 Designeransicht	7
1.4 Programmieren in der Blockansicht	8
1.5 Komponenten	9
1.6 Beispielprojekte	10
1.7 Fazit	21
2 Weitere Möglichkeiten	22
3 Glossar	23
4 Weiterführende Links	25

VORSCHAU

1 App Inventor®

Der MIT App Inventor® ist ein Programm, mit dem auf vergleichsweise einfache Art Android®-Apps entwickelt werden können, die sich auf einem Smartphone installieren lassen.¹ Auch hier wird eine grafische Programmiersprache verwendet, die Scratch® und Calliope nicht unähnlich ist. Eine Weiterentwicklung dieser Umgebung ist unter <https://thinkable.com> auch als kommerzielles Angebot verfügbar, das auch iOS®-Geräte unterstützt.

Inventor® ist komplexer als Calliope oder Scratch®. Mit entsprechend motivierten Kindern ist es aber möglich, hiermit erste Apps zu bauen. Die meisten Grundschulkinder haben vermutlich noch kein eigenes Smartphone, viele haben aber zumindest zu Hause zeitweise Zugriff auf eins. Dadurch ist das Bauen einer App, die nachher auf einem echten Gerät läuft, in gewisser Weise „echter“ als die Browserumgebung von Scratch®.

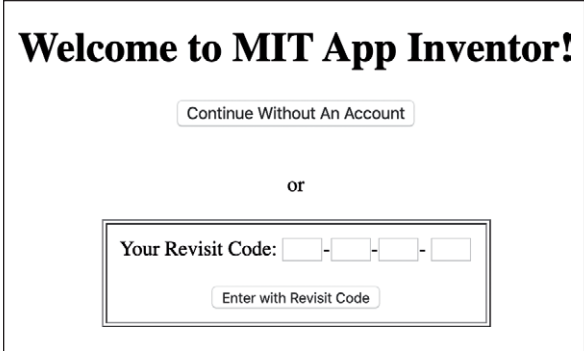
Das technische Set-up für die Ausführung von App-Inventor®-Apps ist etwas komplexer als bei Scratch® oder Calliope, insbesondere was die Installation von Software (siehe Glossar, Seite 23) und die Einrichtung der Netzwerkverbindungen angeht. Dies sollten Sie auf jeden Fall im Vorfeld testen. Aufgrund der zusätzlichen technischen Herausforderungen sollten Sie sich vorher genau überlegen, ob der zeitliche Rahmen und die Zusammensetzung der Klasse ein solches Projekt erlauben. Es gibt auf der Seite einige Anleitungen und vorbereitete Unterrichtseinheiten, es lohnt sich, sich hier vorher einmal umzusehen und einiges auszuprobieren.

1.1 Start und Anmeldung

Um mit der Entwicklung zu beginnen, öffnen Sie in einem Browser die Seite <http://ai2.appinventor.mit.edu/>. Wenn Sie auf „Create Apps“ klicken, können Sie sich mit einem Google®-Account anmelden. Eine Möglichkeit ist, speziell für den Kurs einen Google®-Account zu erstellen, auf dem die Projekte gespeichert werden, es ist jedoch auch möglich, ohne Google®-Account den App Inventor® zu nutzen.

Dafür rufen Sie die Seite <http://code.appinventor.mit.edu/?locale=de> auf. Statt eines Accounts wird hier zur Identifikation ein Buchstabencode verwendet. Wenn Sie „Continue Without An Account“ wählen, wird ein neuer temporärer Account erzeugt und ein „Revisit Code“ generiert, über den Sie beim nächsten Zugriff wieder das Projekt aufrufen können. Notieren Sie diesen Code daher sorgfältig. Im Anschluss können Sie mit einer leeren App starten oder eine der Beispiel-Apps auswählen.

Wenn Sie hier kein Projekt auswählen, gelangen Sie zunächst in die Projektübersicht. In der oberen Navigationsleiste findet man allgemeine Einstel-



Welcome to MIT App Inventor!

Continue Without An Account

or

Your Revisit Code: - - -

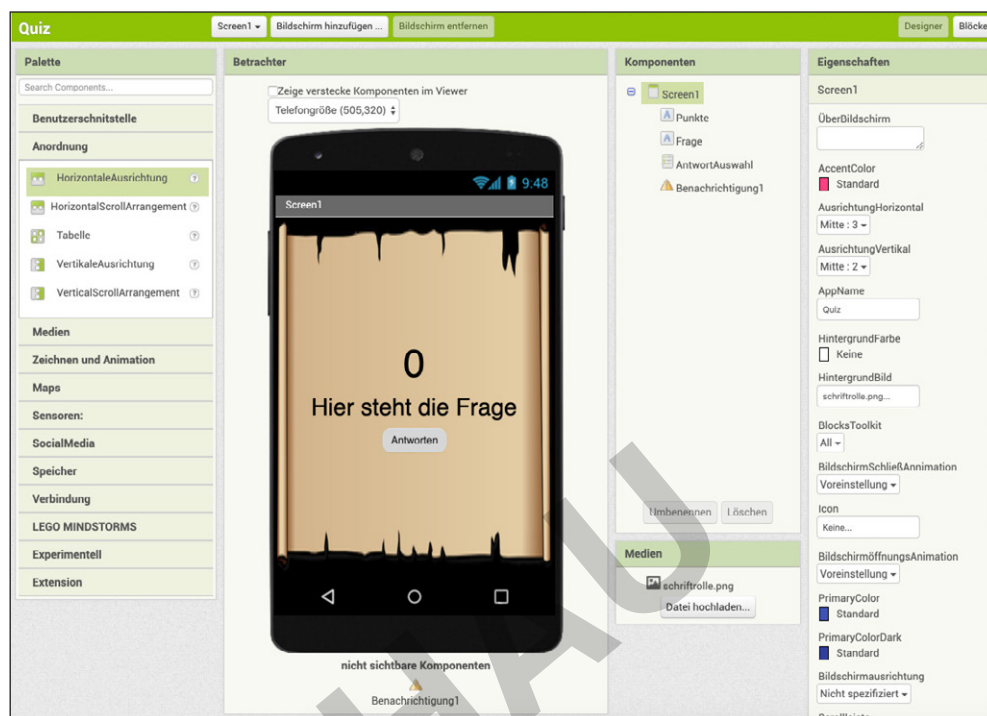
Enter with Revisit Code

App Inventor® ohne Google®-Account nutzen

¹ App Inventor® wurde in Zusammenarbeit mit Google® entwickelt und funktioniert nur mit Android®-Geräten, nicht mit iPhones® und iPads®

1.3 Designeransicht

Das Bauen einer App im App Inventor® besteht aus zwei Teilen: dem Designen der Oberfläche in der Designansicht und dem Programmieren der Interaktionen in der Blockansicht. Die Designansicht besteht aus vier Bereichen:



Designeransicht

Aus der Komponentenpalette werden die verschiedenen verwendeten Programmkomponenten ausgewählt: von Buttons und Textboxen bis hin zu Sensoren und Datenbanken. Diese können per Drag-and-drop in den Betrachter gezogen werden, um sie der App hinzuzufügen.

Im Betrachter können die Komponenten als Vorschau angesehen werden. Über eine Auswahlbox können verschiedene Gerätegrößen ausgewählt werden, um ein Gefühl für das Design auf unterschiedlichen Geräten zu bekommen. Viele Komponenten werden lediglich unter der Betrachteransicht angezeigt, weil sie nicht sichtbar sind, wie etwa eine Musikwiedergabekomponente oder eine Sprachausgabe. Die Komponenten müssen zur App hinzugefügt werden, um in der Programmierung darauf zugreifen zu können.

Der Komponentenbaum zeigt die im Projekt vorhandenen Komponenten in einer Hierarchie an. Komponenten aus dem Bereich „Anordnung“ können andere Komponenten enthalten. Dadurch können Komponenten auch nebeneinander angeordnet werden. Darunter befindet sich der Bereich „Medien“, in dem über den Knopf „Datei hochladen“ Bilder, Videos und Klänge ins Projekt hochgeladen werden können. Diese können dann in den Komponenten referenziert werden, beispielsweise als Hintergrundbild oder für eine Tonwiedergabekomponente.

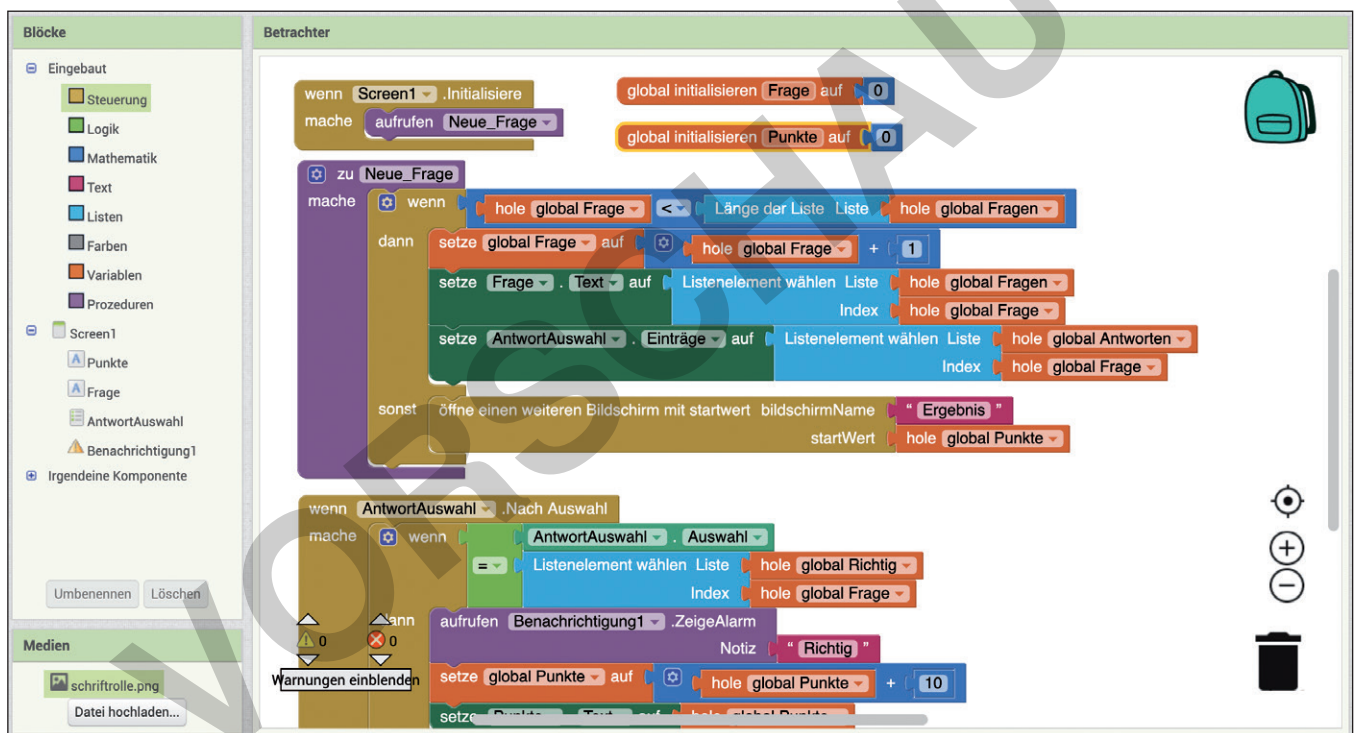
Ganz rechts können die Eigenschaften der Komponenten eingestellt werden. Die auf dem Foto angezeigten Felder beziehen sich immer auf das aktuell ausgewählte Element im Komponentenbaum. Je nach Typ können hier verschiedene Eigenschaften eingestellt werden, die das Erscheinungsbild und Verhalten der Komponente beeinflussen. Teilweise können die Eigenschaften später auch über die Programmierung geändert werden.

Der erste Schritt beim Bauen einer App sollte darin bestehen, das Aussehen der App in der Designeransicht einzustellen. Dabei kann es natürlich sein, dass bestimmte Elemente zum Start der App unsichtbar sein sollen.

Eine App kann auch aus mehreren Bereichen bestehen, die komplett unterschiedlich aussehen. Anstatt ständig Teile der Anwendung unsichtbar zu stellen, können verschiedene Bildschirme gestaltet werden, zwischen denen dann programmatisch umgeschaltet wird.

1.4 Programmieren in der Blockansicht

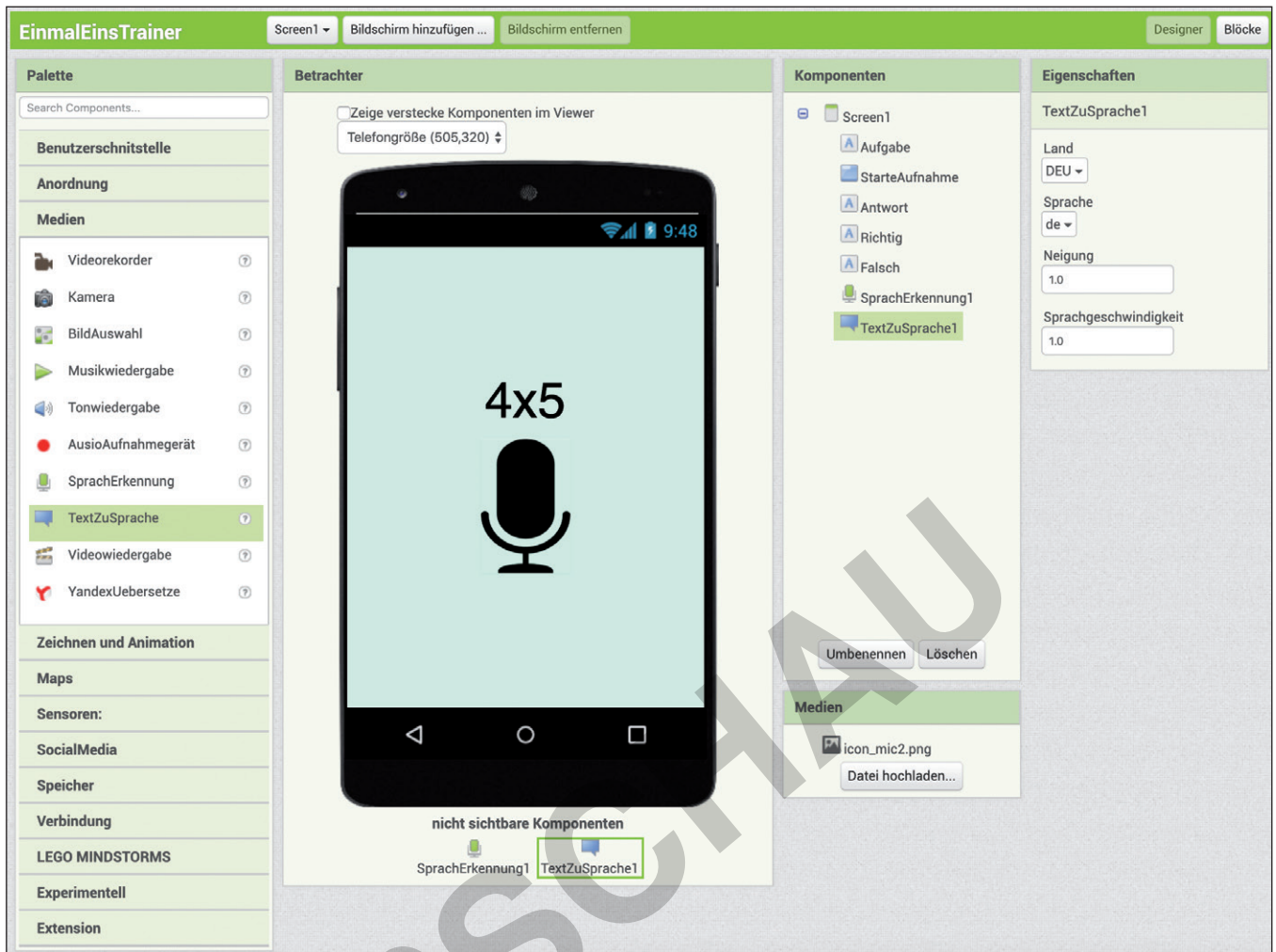
In der Blockansicht wird die Funktion der App programmiert. Über verschiedene Programmblöcke kann hier das Verhalten eingestellt werden, das die App später zeigen soll. Neben Blöcken für normale Programmierkonstrukte wie Variablen, Berechnungen, Logik und Fallunterscheidungen gibt es auch spezielle Blöcke für die App-Komponenten, die in der Designeransicht hinzugefügt worden sind. Wenn man einen Knopf hinzugefügt hat, gibt es also beispielweise einen Block „Wenn Taste1.Klick, mache“, mit dem das Verhalten programmiert wird, wenn dieser Knopf gedrückt wird.



Programmieren in der Blockansicht

Im linken Bereich der Blockansicht finden Sie die Werkzeugpalette. Neben den eingebauten Blöcken findet sich hier auch der Komponentenbaum der App wieder, über den verschiedene Programmblöcke ausgewählt werden können, die sich auf die Komponenten der App beziehen.

Im mittleren Hauptbereich „Betrachter“ können die Programmblöcke frei platziert werden, um unterschiedliche Teilprogramme anzuordnen, die jeweils durch ein Ereignis gestartet werden. Jeder Bildschirm der Anwendung verfügt über ein eigenes Programm, das aus unterschiedlichen Bestandteilen bestehen kann. Über „wenn Screen1.Initialisiere“ können zum Programmstart Ereignisse ausgelöst werden, um die App zu initialisieren, andere Programmteile werden ausgeführt, wenn bestimmte Ereignisse in der App ausgelöst werden, wie etwa das Drücken eines Knopfes oder eine Auswahl in einem Auswahlfeld.

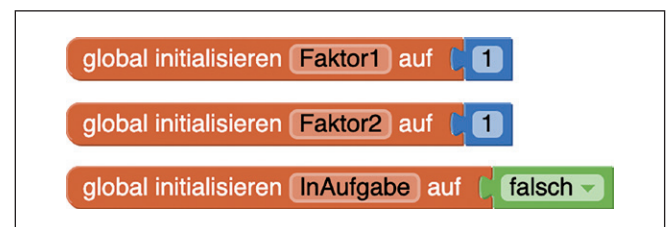


Komponenten der Einmaleins-App

Programmierung

Wechseln Sie über den Schalter „Blöcke“ in die Programmierung. Im Wesentlichen besteht das Programm für diese App aus zwei Teilen: Zum einen wird eine neue Aufgabe generiert und abgefragt, zum anderen wird die eingegebene Antwort überprüft und eine Rückmeldung an den Benutzer gegeben.

Zuerst werden die zwei Variablen „Faktor1“ und „Faktor2“ global definiert, um die aktuell gestellte Aufgabe zu speichern, sowie eine Variable „inAufgabe“, um abzubilden, ob aktuell eine Aufgabe gestellt wurde und beantwortet werden kann.



Variablen definieren

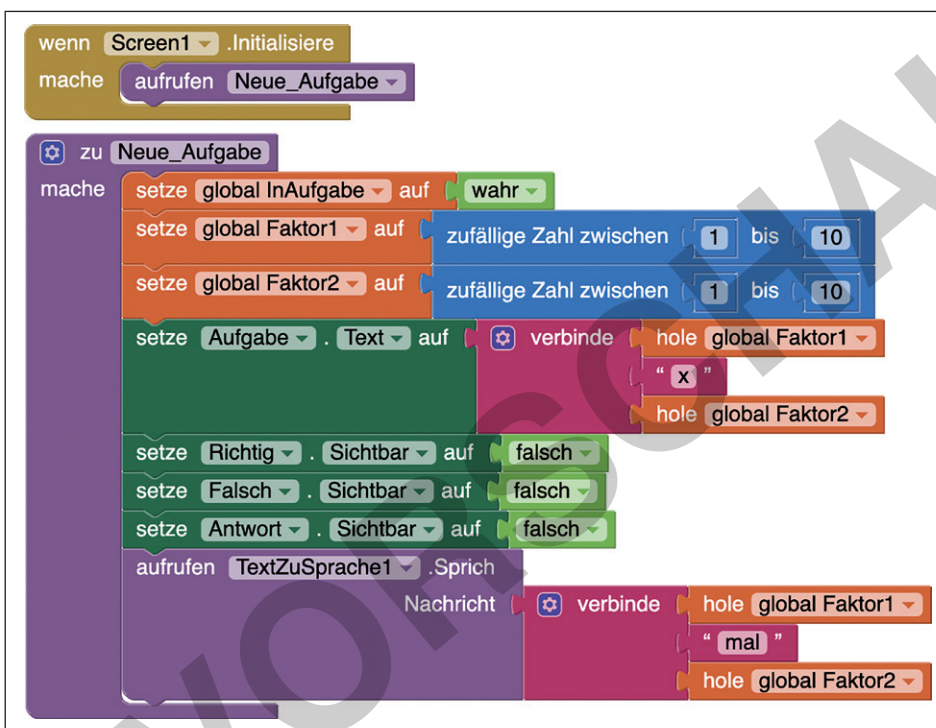
Neue Aufgabe

Um eine neue Aufgabe zu stellen, wird eine Prozedur „Neue Aufgabe“ erstellt, die beim Start der App über „wenn Screen1.Initialisiere“ aufgerufen wird, sowie im weiteren Verlauf, nachdem eine Aufgabe beantwortet wurde. In dieser Prozedur wird eine neue Aufgabe ausgewählt, in der App angezeigt und per Sprachausgabe ausgegeben.

Dafür werden zunächst die Variablen mit neuen Werten versehen. „InAufgabe“ wird auf „wahr“ gesetzt, für die Faktoren wird jeweils eine „zufällige Zahl zwischen 1 und 10“ ermittelt.

Diese Aufgabe wird dann im Feld „Aufgabe“ angezeigt, indem der Wert für „Aufgabe.Text“ gesetzt wird. Mit einem „verbinde“-Block werden die Faktoren und ein Malzeichen verbunden, um den Aufgabentext zusammenzustellen. Die Felder „Richtig“, „Falsch“ und „Antwort“ werden versteckt, indem ihr Wert „Sichtbar“ auf „falsch“ gesetzt wird.

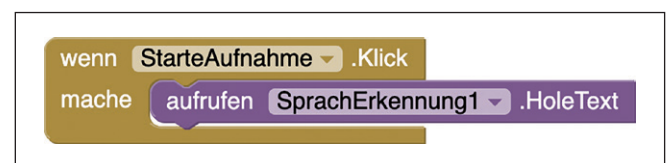
Anschließend soll die Aufgabe auch per Sprachausgabe ausgegeben werden. Hierfür wird der Block „TextZuSprache1.Sprich Nachricht“ verwendet, der bei den Komponenten zu finden ist. Als Eingabe bekommt dieser Block dieselbe verbundene Aufgabe, die im Aufgabenfeld angezeigt wird. Die Sprachausgabe erkennt ein „x“ als „mal“, es kann aber auch als Wort geschrieben werden. Insgesamt ergibt sich für die Aufgabenstellung das folgende Programm:



Prozedur: Neue Aufgabe

Antwort aufnehmen

Um die Aufgabe zu beantworten, kann ein Benutzer auf den Knopf mit dem Mikrofon drücken, um per Spracheingabe die Lösung einzugeben. Das entsprechende Teilprogramm startet mit „wenn StarteAufnahme.Klick“ und ruft die Spracherkennung über den Block „Spracherkennung1.HoleText“ auf.



Aufnahme starten

Wird dieser Block aufgerufen, startet eine Texterkennungskomponente, über die der Benutzer einen Text einsprechen kann. Das Ergebnis wird in einem neuen Block zur Verfügung gestellt, der über ein Ereignis an der Spracherkennungskomponente angesprochen wird.