## Werkzeuge für den Informatikunterricht

# Kara, der programmierbare Marienkäfer

## 1. Kurzvorstellung

Kara ist eine Software, in der der gleichnamige Marienkäfer „lebt“. Es können verschiedene zweidimensionale Welten erstellt werden in denen Kara programmiert werden kann kleine Aufgaben zu erledigen. Die Programme sind dabei endliche Automaten (deterministisch und nichtdeterministisch), die in einer grafischen Entwicklungsumgebung erstellt werden können.
Die gesamte Software ist für den schulischen Einsatz entwickelt worden, weshalb hier einfach und schnell kleine Automaten entwickelt werden können, ohne die Schülerinnen und Schüler mit umfangreichen Entwicklungsumgebungen konfrontieren zu müssen.
Daher bietet Kara den optimalen Einstieg in die Grundlagen der Programmierung sowie in das Modell der endlichen Automaten.

## 2. Einordnung in die Lehrpläne

| Schulart | Klassenstufe | Lernbereich | Kommentar |
|---|---|---|---|
| Oberschule | 8 | LB 2 | Nutzung zur Einführung in Algorithmierung möglich, jedoch besser mit z.B. Scratch. |
| Gymnasium | 8 | LB 2 | |
| | 9/10 | LB 4 | Zur Einführung in die Grundlagen der Programmierung |
| | 11/12 | LB 2 | Modellierung von Automaten |
| Berufsschule | | | Kein sinnvoller Anwendungsbereich, da die theoretische Informatik für die Berufsbildung nicht betrachtet wird. |
| Berufliches Gymnasium | 12/13 | LB 1 | Zur Erschließung des informatischen Modells des endlichen Automaten |
| | | LB 3 | Zur Einführung in die Grundlagen der Programmierung |

## 3. Lernziele

**Kognitive Lernziele:**

Die Schüler-innen lösen einfache Problemstellungen algorithmisch.
Die Schüler-innen lernen die algorithmischen Grundstrukturen Sequenz, Selektion und Zyklus kennen.
Die Schüler-innen lernen das informatische Konzept endlicher Automaten kennen.

**Psychomotorische Lernziele:**

Die Schüler-innen modellieren mithilfe endlicher Automaten einfache Problemlösungen.
Die Schüler-innen implementieren verbal oder schematisch dargestellte Abläufe in einer didaktisch reduzierten Lernumgebung.

**Affektive Lernziele:**

Die Schüler-innen erkennen die Notwendigkeit einer der Implementierung vorgehenden Problemmodellierung.
Die Schüler-innen werden auf die Grenzen der Berechenbarkeit aufmerksam.

## 4. Kompetenzentwicklung

**Fachkompetenz:**

Die Schüler-innen sind in der Lage endliche Automaten zu analysieren.
Die Schüler-innen können Automaten zustandsorientiert modellieren.

**Lern-/Methodenkompetenz:**

Die Schüler-innen sind in der Lage Algorithmen mit den algorithmischen Grundbausteinen zu entwerfen und diese geeignet darzustellen.
Die Schüler-innen können modellierte Automaten in einer Programmierumgebung implementieren.

**Sozialkompetenz:**

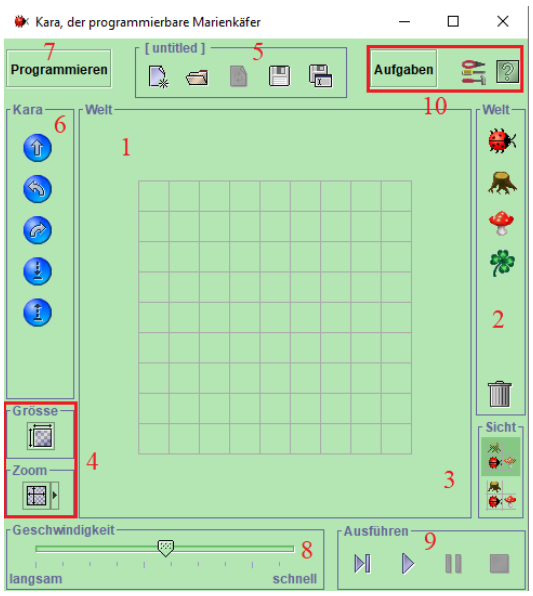
Die Schüler-innen können sich mit anderen über verschiedene Wege der Problemlösung austauschen.
Die Schüler-innen können verschiedene Lösungsansätze eines Problems nachvollziehen und Vor- und Nachteile mit den Urhebern der Ansätze diskutieren.

**Selbstkompetenz:**

Die Schüler-innen sind in der Lage ihren Lösungsweg zu reflektieren und nach besseren Alternativen zu suchen.
Die Schüler-innen können ihre Lösung eines Problems selbstständig an verschiedenen Ausgangssituationen testen und verbessern.

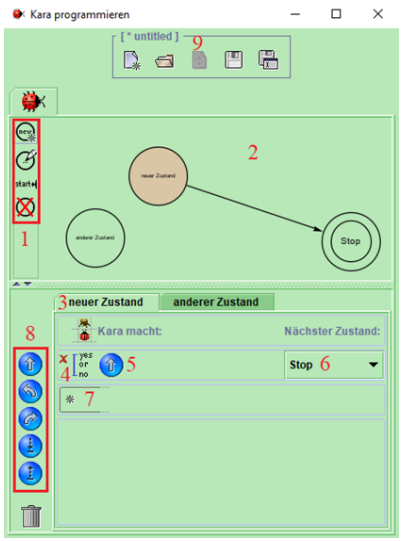
## 5. Prinzipieller Aufbau

**Hauptfenster**




1 Welteditor – hier können Welten erstellt und bearbeitet werden. Ebenso wird hier die Abarbeitung der Programme durch Kara angezeigt
2 Um Elemente in der Welt zu platzieren, werden sie von hier in den Welteditor per Drag and Drop hineingezogen
3 Hier kann eingestellt werden, ob die Objekte, die Kara mit seinen Sensoren aktuell registrieren kann hervorgehoben werden sollen.
4 Hier können die Dimensionen des Weltfensters sowie der Zoom festgelegt werden.
5 Hier kann man eine neue Welt erstellen, eine vorhandene öffnen, sowie die aktuelle Welt neu laden und speichern.
6 Mit den Knöpfen kann Kara manuell gesteuert werden.
7 Hier gelangt man in das Programmierfenster.
8 Hier kann man die Abarbeitungsgeschwindigkeit des Programms einstellen.
9 Hier gibt es Knöpfe zum Starten, Pausieren und Stoppen des Programms. Mit dem linken Knopf kann das Programm außerdem schritt für Schritt abgearbeitet werden.
10 Hier finden sich vorgefertigte Aufgaben sowie kleinere Einstellungen (Beispielsweise von deterministischen Automaten auf nichtdeterministische umstellen) und eine Hilfe Funktion.

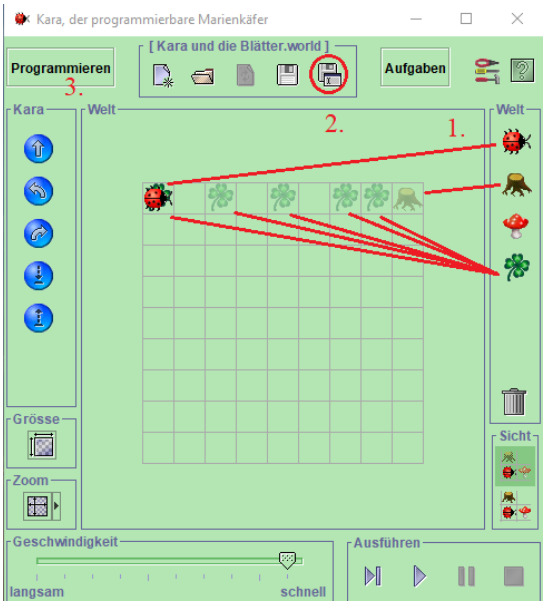
**Programmierfenster**




1 Neuen Zustand hinzufügen, vorhandenen Bearbeiten, zum Startzustand machen oder löschen.
2 Aktueller Zustandsgraph des Automaten
3 Zustandsübergänge eines Zustandes
4 Auswahl der Sensorzustände (Eingabe), die für diesen Übergang herrschen müssen.
5 Auswahl der Aktionen, die Kara in diesem Übergang ausführen soll.
6 Auswahl in welchen Zustand Kara übergehen soll.
7 Neuen Zustandsübergang hinzufügen.
8 Von hier können Aktionen, die Kara ausführen soll nach *5* gezogen werden.
9 Neues Programm, vorhandenes Programm öffnen, neu laden und speichern

## 6. Handhabung

**Welt erstellen**




- Welt bauen, in der sich Kara bewegen kann aus
  - Baumstämmen, (unbeweglich, Kara hat Sensoren für „Baum vorne“, „Baum links“ und „Baum rechts“)
  - Pilzen, (können von Kara verschoben werden, Sensor für „Pilz vorne“)
  - Kleeblättern (können von Kara aufgenommen und hingelegt werden, Sensor für „Kleeblatt unten“)
  - und Kara selbst
- Dafür Elemente per Drag and Drop auf das Raster ziehen (1.)
- Kara muss so platziert werden, wie er im Startzustand stehen soll (eventuell noch manuell drehen mit den Bedienelementen auf der linken Seite)
- Welt speichern (2.)
- In dieser Welt kann Kara dann über die Buttons links gesteuert werden oder wie im nächsten Schritt gezeigt programmiert werden

**Kara programmieren**

Dafür muss man bereits das Programmierfenster über das Hauptfenster geöffnet haben mit Programmieren.

- Neuen Zustand erstellen mit → neues Fenster öffnet sich
- Sensoren (Eingabealphabet) für diesen Zustand wählen und diese per Drag and Drop aus der Sensorbibliothek rechts nach links zu den benutzten Sensoren ziehen
- Sensorauswahl bestätigen mit
- Für alle möglichen Kombinationen aus Sensorzuständen festlegen, was Kara tun soll
  - Bewegungen (Schritt gehen, links drehen, rechts drehen, Kleeblatt ablegen und Kleeblatt

aufnehmen) aus der Bewegungsauswahl (siehe Prinzipieller Aufbau Programmierfenster *8*) per Drag and Drop zur *5* ziehen
    - Auswählen, in welchem Zustand sich Kara danach befinden soll
- Beliebig viele Zustände hinzufügen
- Startzustand festlegen mit 

- Überprüfen, ob der Endzustand erreicht wird
- Programm speichern, Programmierfenster offenlassen

Dann kann das Programm über  gestartet werden. Außerdem kann man über

-  die Programmausführung pausiert werden,
-  das Programm in einzelnen Schritten durchlaufen werden,
-  die Programmausführung beendet werden
- 

die Abarbeitungsgeschwindigkeit eingestellt werden

**Fertige Aufgaben verwenden**

- Über *10* im Hauptfenster „Aufgaben“ auswählen
- Über Dropdown-Menü beliebige Aufgabe auswählen
- Beschreibung der Aufgabe durchlesen und wie oben beschrieben Kara programmieren
- Wenn fertig programmiert über den Reiter „Welten“ im Aufgabenfenster eine Welt auswählen und darin das Programm, wie oben laufen lassen
    - Einige Aufgaben haben die Möglichkeit das Programm automatisch in mehreren Welten prüfen zu lassen – dazu dann einfach den „Überprüfen“ Knopf verwenden
- Musterlösung lässt sich im Reiter „Lösung“ ansehen und Programm über „Programm anzeigen“ in das Programmierfenster laden
    - Hinweis: Es gibt auch eine Programmversion, in der die Lösungen nicht vorhanden sind

**Einstellungen vornehmen**

- Über  im Hauptfenster das Einstellungsfenster öffnen
- Einstellungsmöglichkeiten:
    - Farbthema grün, blau oder weiß – weiß praktisch für Screenshots für z.B. Arbeitsblätter
    - deterministische oder nichtdeterministische Ausführung der programmierten Automaten einstellen
    - Statistiken über die Programme anzeigen lassen

## 7. Screencast

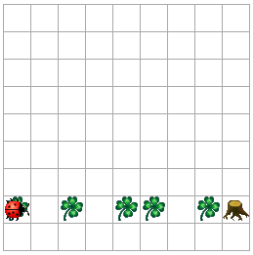
Hinter diesem Link verbirgt sich ein kurzes Video, in dem die Erstellung einer einfachen Welt und die Programmierung Karas an einem Beispiel gezeigt werden.

https://t1p.de/ddi-karascreencast

## Übungsblatt Kara, der programmierbare Marienkäfer

### Aufgabe 1

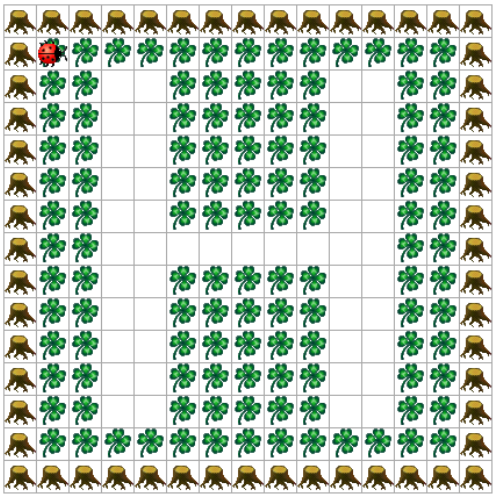
**10 min**



a Baue die Welt auf der rechten Seite im Welteditor nach.
b Speichere die soeben von dir erstellte Welt als Aufgabe_1.world ab.
c Kara hat Hunger! Programmiere ihn so, dass Kara alle Kleeblätter vor ihm essen (aufnehmen) kann, bis er beim Baumstumpf angekommen ist. Speichere dieses Programm als Aufgabe_1.kara

### Aufgabe 2

**15 min**



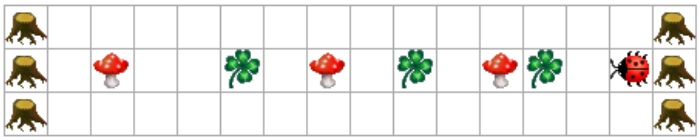
In Karas Wald gibt es einen Landeplatz für Marienkäfer, die nachts von der Kleeblattsuche zurückkommen. Um ihn besser erkenntlich zu machen wurde der Platz mit Kleeblättern ausgelegt, die nachts leuchten und nur ein „H“ ähnlich, wie bei Hubschrauberlandeplätzen wurde freigelassen.
Kara findet es allerdings schade um die ganzen wertvollen Kleeblätter und hat bei der Verwaltung nachgefragt, ob man das nicht etwas weniger verschwenderisch umsetzen könnte.

a Beschreibe, wie man den Landeplatz mit weniger Kleeblättern gestalten könnte, sodass das „H“ aber in derselben Größe zu sehen bleibt. Die Umsetzung der Umgestaltung muss in einem Programm beschreibbar sein, um von Kara durchgeführt werden zu können.
b Öffne `Aufgabe_2.world` und schreibe ein Programm, in dem Kara den Platz umgestaltet. Beachte dabei, dass das Programm für verschiedene Muster funktionieren soll und, dass Kara immer oben links startet. Speichere dein Programm unter `Aufgabe_2.kara` ab.
c Gibt an, wie viele leuchtende Kleeblätter durch die Umgestaltung gespart wurden.

### Aufgabe 3

**15 min**

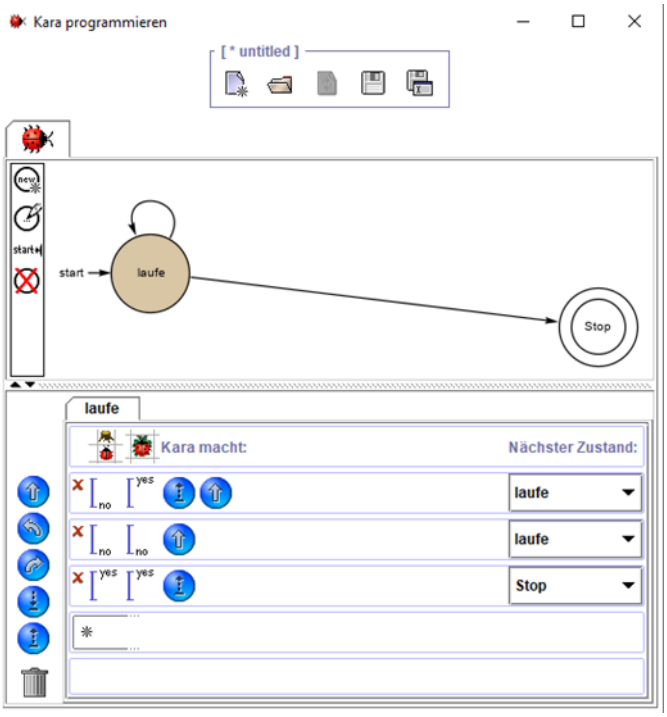
Kara hat sich in seinem Wald eine kleine Sammlung an Kleeblättern angelegt, damit er sich nicht jeden Tag auf die Suche nach neuen Kleeblättern begeben muss, sondern auch mal einen Tag zuhause bleiben kann.
Sein Bruder hat ihm allerdings das letzte Mal einige seiner Kleeblätter gestohlen und nun möchte Kara sie vor ihm verstecken. Dazu hat er alle Kleeblätter in der Nähe von Pilzen abgelegt und möchte sie nun damit bedecken.



a Öffne die Weltdatei `Aufgabe_3.world`.
b Programmiere Kara so, dass er alle Pilze auf die Kleeblätter jeweils rechts von ihnen schiebt und die Kleeblätter so versteckt. (siehe Bild)
Hinweis: Denke daran, dass dein Programm auch für andere Welten funktionieren soll, in denen die Pilze und Kleeblätter genauso angeordnet sind!
c Speichere dein Programm unter `Aufgabe_3.kara` ab.

## Lösungen Übungsblatt

### Aufgabe 1




### Aufgabe 2

a) Mehrere Möglichkeiten:
Freie Fläche mit Kleeblättern auslegen (am einfachsten)
Umrandung stehen lassen (schwierig)

b)



c) $120 - 49 = 71$

**Aufgabe 3**


suchePilz
schiebePilz
Kara macht:
Nächster Zustand:
no
no
suchePilz
yes
no
schiebePilz
no
yes
Stop


schiebePilz
suchePilz
Kara macht:
Nächster Zustand:
no
schiebePilz
yes
suchePilz