



UNIVERSITÄT
LEIPZIG

Didaktik der Informatik

Lehramt Informatik (Gy, OS, BS, FS)
Modul Didaktik der Informatik - E-Learning und Tools

Werkzeuge für den Informatikunterricht

---- PythonKara----

Empfohlen für Klasse: 8

Autor: Anabel Wensky

Lizenz: Dieses Werk ist lizenziert unter [CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)



1. Kurzvorstellung

PythonKara ist eine kostenfreie, plattformübergreifende Lernsoftware, die eine Einführung in die Grundlagen der Algorithmik und Programmierung bietet. Die didaktisch reduzierte Lernumgebung von PythonKara ermöglicht es dem Benutzer, in einer schachbrettartigen Welt den Marienkäfer Kara zu steuern und durch Programmierung unterschiedliche Tätigkeiten ausführen zu lassen. Diese Software unterstützt mehrere populäre Programmiersprachen, darunter Python, Ruby, JavaScript und Java, wobei der Schwerpunkt in dieser Darstellung auf der Programmierung mit Python liegt.

2. Einordnung in den Lehrplan

Gymnasium	Klasse 8: Lernbereich 1: „Algorithmen“ In diesem Lernbereich geht es um das Beherrschen der algorithmischen Lösung einfacher Problemstellungen in einer didaktisch reduzierten Programmierumgebung. Strukturen wie die Sequenz, Verzweigung und Wiederholung sollen von den Lernenden umgesetzt werden können. Mit PythonKara können diese Ziele erfüllt werden. Dabei können die Schülerinnen und Schüler den Algorithmus selbstständig auf Richtigkeit überprüfen.
	Klasse 10: Lernbereich 1: „Algorithmen“ In diesem Lernbereich geht es um das Beherrschen der Implementierung der algorithmischen Grundstrukturen wie Datentypen, Variablenzuweisung und verknüpfte Bedingungen. PythonKara bietet Aufgaben für jede Lernstufe an. Auch diese Lernziele können mit der Lernsoftware PythonKara abgedeckt werden.
	Grundkurs Jahrgangstufen 11/12 Lernbereich 2: „Algorithmmierung und Programmierung“ Beherrschen der Implementierung strukturierter Datentypen in einer Programmierumgebung, das Kennen von Konzepten der objektorientierten Programmierung und das Kennen der Rekursion und Iteration sind Inhalte des Lernbereichs und lassen sich in PythonKara umsetzen.

Auch im Leistungskurs kann diese Software genutzt werden. Für fortgeschrittenere Aufgaben wird jedoch empfohlen eventuell eine andere IDE zu nutzen, weshalb hier eine klare Empfehlung für die Nutzung in Klassenstufe 8-11 Grundkurs ausgesprochen wird.

Oberschule

Klasse 8: Lernbereich 1: „Algorithmen und Programme“

Die Schüler modellieren Lösungen zu einfachen Problemen mit Hilfe von verschiedenen Darstellungen eines Algorithmus und implementieren diese in einer didaktisch reduzierten Programmierumgebung. Dabei nutzen sie die Phasen des Problemlöseprozesses. In diesem Lernbereich übertragen die Lernenden ihr Wissen über algorithmischen Grundstrukturen in die Entwicklungsumgebung und erhalten eine visuelle Rückmeldung.

3. Lernziele

Affektive Lernziele:

Die Lernenden begreifen die Relevanz von Algorithmen, Programmiersprachen und grundlegenden algorithmischen Strukturen für die Lösung von Anwendungsproblemen.

Die Lernenden arbeiten kooperativ mit ihren Mitschülern zusammen, teilen ihre Lösungsansätze und diskutieren unterschiedliche Methoden, um die Effektivität ihrer Algorithmen zu verbessern.

Die Lernenden gewinnen Vertrauen in ihre Fähigkeiten, algorithmische Problemstellungen zu analysieren und zu lösen, und fühlen sich sicher dabei, eigene Lösungen zu entwickeln und zu überprüfen.

Kognitive Lernziele:

Die Lernenden können die grundlegenden Konzepte der Sequenz, Verzweigung und Wiederholung erklären und anwenden.

Die Lernenden können einfache Problemstellungen analysieren und passende algorithmische Lösungen entwerfen.

Die Lernenden können typische Fehler in ihren Programmen identifizieren und geeignete Korrekturen vornehmen.

Lernende beherrschen die Grundbegriffe der Algorithmik und Programmierung.

Psychomotorische Lernziele:

Die Lernenden beherrschen Drag & Drop.

Die Lernenden können Programmcode korrekt und effizient in eine Programmierumgebung eingeben.

Die Lernenden skizzieren algorithmische Lösungswege und überführen diese in funktionierenden Programmcode.

4. Kompetenzentwicklung

Fachkompetenz

Die Lernenden erwerben die Fähigkeit, grundlegende algorithmische Konzepte wie Sequenzen, Verzweigungen und Schleifen zu verstehen und anzuwenden.

Die Lernenden entwickeln Fertigkeiten im Schreiben und Debuggen von Programmen.

Die Lernenden erlangen ein tiefes Verständnis für die Syntax und Semantik der Programmiersprache Python und können deren Werkzeuge und effektiv nutzen.

Lern- und Methodenkompetenz

Die Lernenden entwickeln systematische Ansätze zur Problemlösung, indem sie Problemstellungen analysieren, Lösungen entwerfen, implementieren und testen.

Die Schülerinnen und Schüler sind in der Lage, Fehler zu identifizieren und zu korrigieren, wodurch sie ihre analytischen und methodischen Fähigkeiten stärken.

Die Lernenden sind in der Lage, sich eigenständig in neue Themenbereiche einzuarbeiten, Informationsquellen zu nutzen und ihre Lernprozesse zu strukturieren.

Sozialkompetenz

Die Lernenden arbeiten in Gruppen an Projekten, tauschen Ideen aus und unterstützen sich gegenseitig bei der Lösung von Problemen.

Die Lernenden sind in der Lage, effektiv zu kommunizieren und gemeinsam Lösungen zu erarbeiten, was ihre Teamfähigkeit und Kommunikationskompetenz stärkt.

Selbstkompetenz

Durch das Lösen komplexer Aufgaben und das Bewältigen von Herausforderungen stärken die Lernenden ihr Selbstbewusstsein und ihre Widerstandsfähigkeit.

Die Schülerinnen und Schüler sind in der Lage, aus Fehlern zu lernen und sich nicht entmutigen zu lassen.

3. Handhabung

3.1 Installation

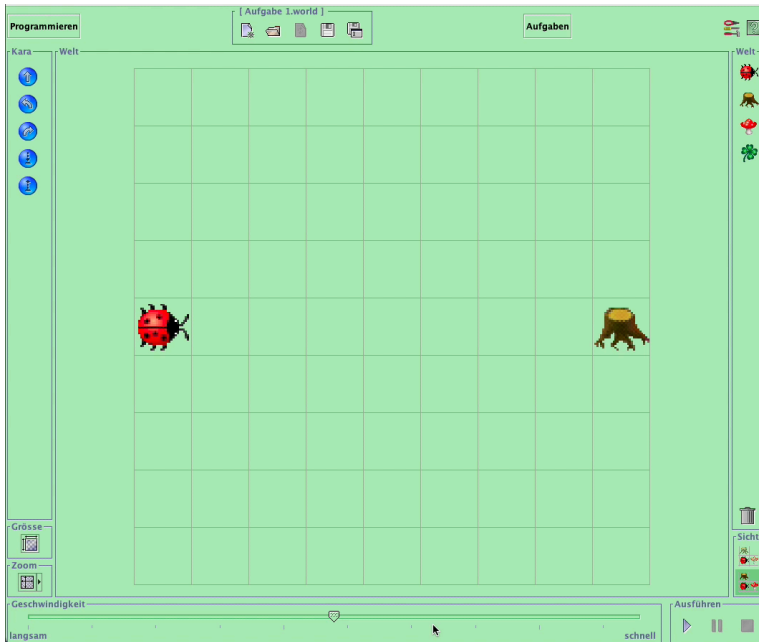
Die Lernsoftware findest du unter:

<https://www.swisseduc.ch/informatik/karatojava/pythonkara/>

Es gibt 2 Versionen. Achte darauf, dass die Version **ohne** Lösungen für Schüler und Schülerinnen installiert wird.

3.2 Öffnen, Speichern und Schließen

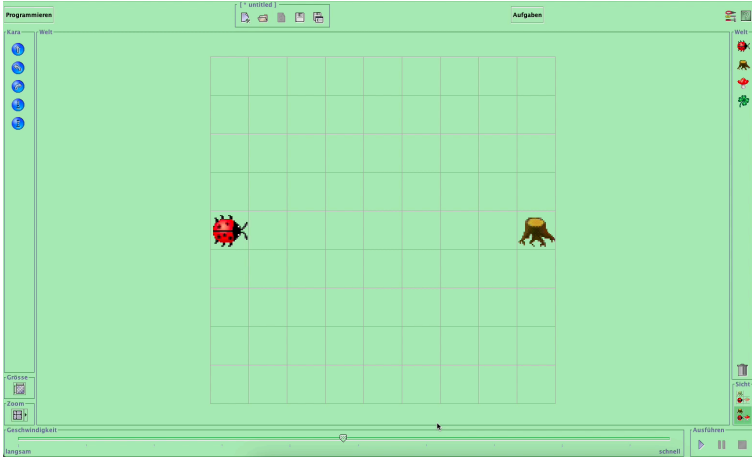
In der Statusleiste, welche sich oben befindet, kann ein Programm geöffnet, erstellt oder gespeichert werden.



Hinweis: Unter „Aufgaben“ findest du bereits vorhandene Welten und Aufgaben auf einem höheren Niveau. Diese Manual bietet einen Einstieg und Aufgaben für Lernende, welche zuvor noch nie programmiert haben.

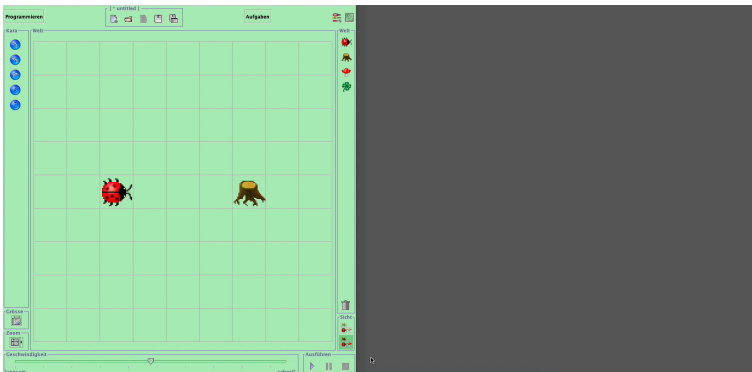
3.3 Hilfsmittel und Übersicht

In den Einstellungen (Fragezeichenfeld) befindet sich die Bedienungsanleitung für die Lernsoftware. Dort findest du eine Übersicht, Methoden und die Objektbeschreibung.



3.4 Programmieren

Es existiert ein Feld „Programmieren“. Wenn es angeklickt wird, öffnet sich ein neues Fenster, indem die Schülerinnen und Schüler ihr Programm schreiben. Wird unten rechts auf „Ausführen“ geklickt, wird das Programm ausgeführt.

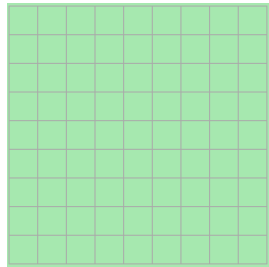


Aufgaben

Für jede Aufgabe gibt es ein Bild, welches die Welt von Kara zeigt. Dieses Bild musst du in deine Welt übertragen. Denke daran, deine Lösungen ab Aufgabe 1 zu speichern.

0. Erkunde die Welt von Kara

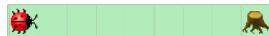
1. Füge Kara in die Welt ein
2. Füge ein Kleeblatt ein
3. Lass Kara zum Kleeblatt laufen
4. Kara soll das Kleeblatt aufsammeln
5. Füge einen Pilz ein.
6. Lass Kara gegen den Pilz laufen
7. Lege ein Kleeblatt ab
8. Füge einen Baum ein
9. Lass Kara gegen den Baum laufen
10. Entferne alle Elemente



Falls du dir unsicher bist, wie du Kara ohne zu programmieren steuern kannst, schau dir die Informationen in den Einstellungen noch einmal an. Du bist fertig? Dann entferne die Elemente, indem du sie zum Mülleimer ziehst und dort lässt.

1. Kara und der Baum

Schreibe ein Programm, welches Kara zum nächsten Baum führt.



Hinweis: Gehe in den Einstellungen auf "PythonKaraProgramm" und suche nach einer Methode, welche Kara einen Schritt nach vorne gehen lässt. Schreibe im Programmierfenster "kara." vor die Methode, wenn Kara diese ausführen soll. Beispiel: kara.turnLeft()

Benutze folgende Methoden: move()

2. Kara und die Kleeblätter (While-Schleife)

Schreibe ein Programm, welches Kara nur auf Blättern laufen lässt. Benutze dafür die While-Schleife.

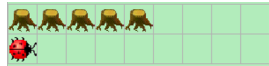


*Hinweis: Gehe in den Einstellungen auf "PythonKaraProgramm" und suche nach einer Bedingung, welche überprüft: "Kara auf Kleeblatt?". Schreibe im Programmierfenster while + "kara." vor die Bedingung und setze danach einen Doppelpunkt. Mit einer While-Schleife gestaltest du dein Programm effizienter. Anstatt 9 Zeilen wirst du nur 2 benötigen. Beispiel: while kara.mushroomFront():
kara.move()*

Benutze folgende Methoden: move(), onLeaf()

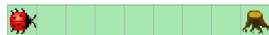
3. Kara unter den Bäumen (While-Schleife)

Schreibe ein Programm, welches Kara nur unter den Bäumen laufen lässt. (bis links von Kara kein Baum mehr existiert)



4. Optimierte dein Programm: Kara und der Baum

Schreibe ein Programm, welches Kara zum nächsten Baum führt. Benutze die While-Schleife.



Hinweis: Die While-Schleife kann mit dem Wort "not" erweitert werden. (Solange etwas nicht erfüllt ist)

*Beispiel: while not kara.onLeaf():
kara.move()*

Benutze folgende Methoden: move(), treeFront()

5. Kara und die Blätter (if-Anweisung)

Erweitere dein Programm mit einer if-Anweisung. Kara soll bis zum nächsten Baum geführt werden. Wenn auf dem Feld ein Blatt liegt, soll Kara es aufnehmen.



Hinweis: Schreibe in deine While-Schleife eine if-Anweisung. Benutze dafür das Schlüsselwort if + Bedingung + Doppelpunkt.

Beispiel: while not kara.mushroomFront():
if kara.treeFront():
kara.turnLeft()
kara.move()

Benutze folgende Methoden: move(), treeFront(), onLeaf(), removeLeaf()

6. Kara und die Blätter (if-else-Anweisung)

Erweitere dein Programm: Füge in die if-Anweisung das Schlüsselwort "else" hinzu. Wenn auf dem Feld ein Blatt liegt, soll Kara es aufnehmen; falls nicht, soll Kara ein Blatt ablegen.



Hinweis: if - else-Anweisung (Wenn Fall 1 eintritt "mach das", sonst "mache das andere")

Beispiel: while not kara.mushroomFront():
if kara.treeFront():
kara.turnLeft()
else:
kara.turnRight()
kara.move()

Viel Erfolg!

Lösungen

Aufgabe 1

7x kara.move() untereinander

Aufgabe 2

```
while kara.onLeaf():  
    kara.move()
```

Aufgabe 3

```
while kara.treeLeft():  
    kara.move
```

Aufgabe 4

```
while not kara.treeFront():  
    kara.move()
```

Aufgabe 5

```
while not kara.treeFront():  
    if kara.onLeaf():  
        kara.removeLeaf()  
    kara.move()
```

Aufgabe 6

```
while not kara.treeFront():  
    if kara.onLeaf():  
        kara.removeLeaf()  
    else:  
        kara.putLeaf()  
    kara.move()
```

Hinweis: Eine häufige Fehlerquelle ist die falsche Einrückung des Codes. Diese Aufgaben sind die Voraussetzung für die Nutzung der vorhandenen Aufgaben von PythonKara, welche etwas anspruchsvoller beginnen.