

Pythonkurs mit Robot Karol Online

Eine anschauliche und betont nicht mathematiklastige Einführung in die Algorithmik

Peter Ehrlich

peter.ehrlich@franziskaneum.lernsax.de

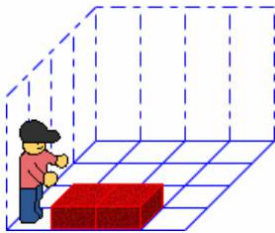
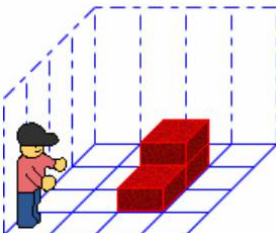
Juli 2024

1 einfache Programme – Sequenz

Karol kennt folgende Befehle

Befehl	Bedeutung
schritt()	Geht einen Schritt vorwärts (wenn das möglich ist!)
linksDrehen()	Dreht sich links um
rechtsDrehen()	Dreht sich rechts um
hinlegen()	Legt einen Ziegel in das Feld vor sich (wenn das möglich ist!)
aufheben()	Hebt einen Ziegel vom Feld vor sich auf (wenn ein Ziegel dort liegt!)
markeSetzen()	Markiert das Feld, auf dem er steht
markeLöschen()	Löscht die Marke vom Feld, auf dem er steht

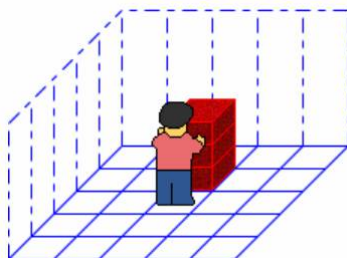
Beispielprogramm <https://karol.arrrg.de/#TX7Z>

Quelltext	Struktogramm	Start	Ziel
<pre> karol = Robot() karol.aufheben() karol.schritt() karol.aufheben() karol.schritt() karol.linksDrehen() karol.hinlegen() karol.schritt() karol.hinlegen(2) karol.schritt() karol.rechtsDrehen() </pre>	<pre> graph TD A[aufheben()] --> B[schritt()] B --> C[aufheben()] C --> D[schritt()] D --> E[linksDrehen()] E --> F[hinlegen()] F --> G[schritt()] G --> H[hinlegen(2)] H --> I[schritt()] I --> J[rechtsDrehen()] </pre>		

Aufgaben

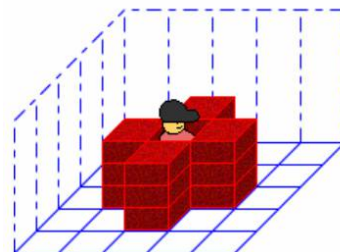
Sequenz-a) <https://karol.arrrg.de/#2MWZ>

Karol bewegt sich in die Mitte des Raumes und baut einen **3-er Stapel**.



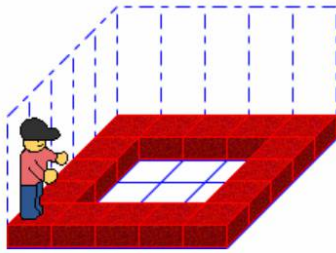
Sequenz-b) <https://karol.arrrg.de/#KB9P>

Karol bewegt sich in die Mitte des Raumes und **mauert sich ein**.



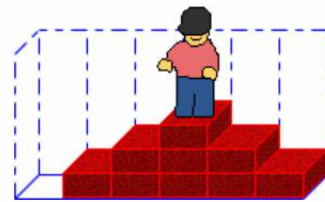
Sequenz-c) <https://karol.rrrg.de/#YE48>

Karol legt in der Welt (5*5) außen ein **Quadrat**.



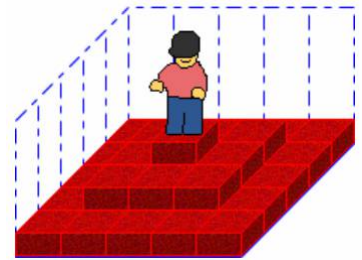
Sequenz-d) <https://karol.rrrg.de/#EBHP>

Karol baut eine **Treppe**.



Sequenz-Zusatz) <https://karol.rrrg.de/#D8KB>

Karol baut in der Welt (5*5) eine **Pyramide**.



2 Wiederholungen

2.1 Zählschleife mit for-Anweisung

Beispielprogramm: Zählschleife – einfach <https://karol.arrrg.de/#SE3H>

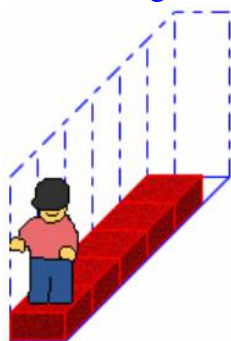
Quelltext	Struktogramm	Start	Ziel
<pre>karol = Robot() # Vorbereitung karol.linksDrehen() karol.schritt() karol.rechtsDrehen() karol.schritt() karol.schritt() # Wiederholung for i in range(4): karol.hinlegen() karol.linksDrehen()</pre>			

Aufgaben

Schreibe je ein Programm, das folgende Ziegelanordnungen erzeugt. Verwende die **Wiederholung** und möglichst wenig den Befehl **Hinlegen**.

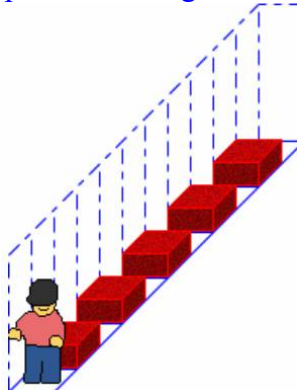
Zählschleife-a)

<https://karol.arrrg.de/#PCRB>



Zählschleife-b)

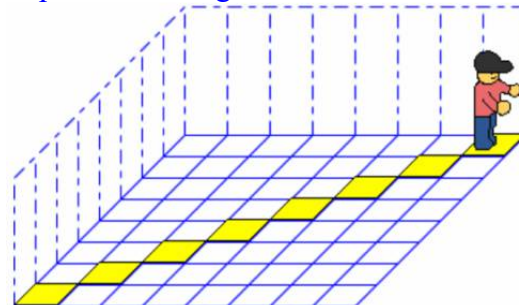
<https://karol.arrrg.de/#3P8A>



Lass die gesamte **Diagonale** einer quadratischen Welt (8*8) mit Markierungen belegen.

Zählschleife-c)

<https://karol.arrrg.de/#YBX3>



Zählschleife-d) <https://karol.arrrg.de/#R8BU>

Was baut Karol bei den folgenden Struktogrammen? Überprüfe Deine Vermutung!

wiederhole 4 mal
hinlegen()
linksDrehen()
schritt()

wiederhole 4 mal
hinlegen()
linksDrehen()
schritt()

wiederhole 4 mal
hinlegen()
linksDrehen()
schritt()

Quelltext	Struktogramm	Ziel
<pre> karol = Robot() # Vorbereitung karol.linksDrehen() karol.schritt() karol.rechtsDrehen() karol.schritt() # äußere Schleife für alle 3 Blumen for j in range(3): # innere Schleife für eine Blume for i in range(4): karol.hinlegen() karol.linksDrehen() # Vorbereitung der nächsten Blume karol.schritt(4) </pre>	<pre> graph TD A[wiederhole 3 mal] --> B[linksDrehen()] A --> C[schritt()] A --> D[rechtsDrehen()] A --> E[schritt()] A --> F[wiederhole 4 mal] F --> G[hinlegen()] F --> H[linksDrehen()] F --> I[schritt(4)] </pre>	

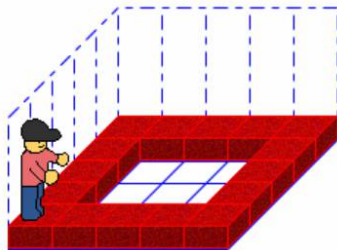
Aufgaben

Zählschleife-e)

<https://karol.arrrg.de/#XV6D>

Karol legt in der Welt (5*5) außen ein **Quadrat**.

Benutze **geschachtelte Wiederholungen** und somit möglichst wenig Befehle.

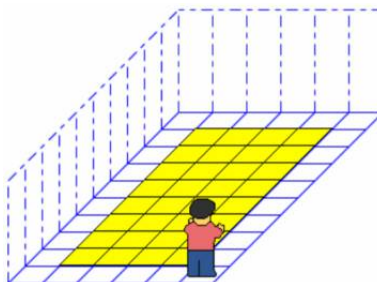


Zählschleife-f)

<https://karol.arrrg.de/#V26Z>

Lege in der gesamten Welt (6*10) ein **Rechteck (4*8) mit Marken** aus.

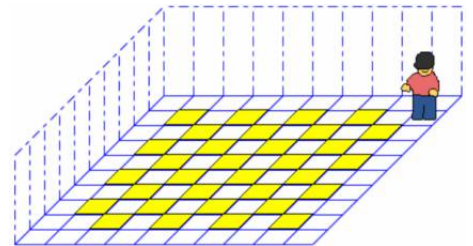
Benutze **geschachtelte Wiederholungen** und somit möglichst wenig Befehle.
Tipp: Lege jeweils eine Zeile aus und begib Dich zurück. Bearbeite so alle 4 Zeilen.



Zählschleife-Zusatz)

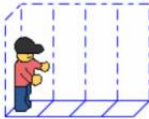
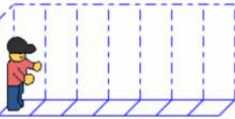
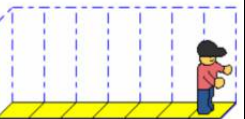
<https://karol.arrrg.de/#7CBG>

Erzeuge in der Welt (10*10) ein **Schachbrettmuster (8*8)** aus Marken.



2.2 Bedingte Wiederholung mit while-Anweisung

Beispielprogramm: bedingte Wiederholung <https://karol.arrrg.de/#NYDS>

Quelltext	Struktogramm	Start	Ziel
<pre>karol = Robot() karol.markeSetzen() while karol.nichtIstWand(): karol.schritt() karol.markeSetzen()</pre>	<pre> graph TD A[markeSetzen()] --> B[wiederhole solange nichtIstWand()] B --> C[schritt()] C --> D[markeSetzen()] D --> B</pre>	 	 

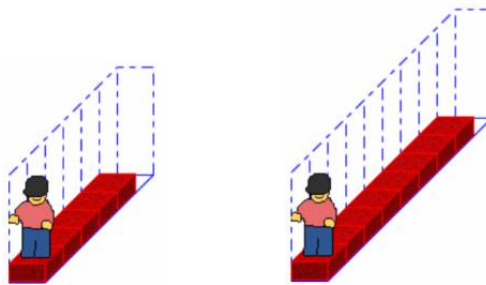
folgende Bedingungen können benutzt werden

istWand	nichtIstWand
istZiegel	nichtIstZiegel
istMarke	nichtIstMarke
istNorden	nichtIstNorden

Aufgaben

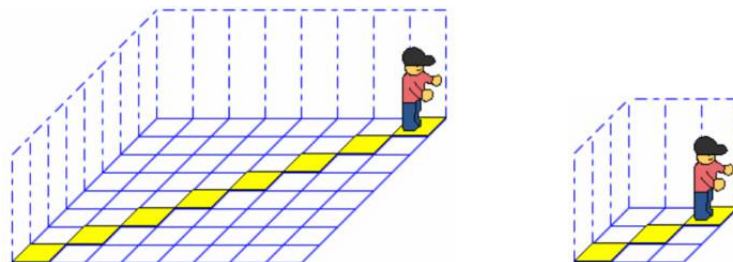
bedingte Wiederholung-a) <https://karol.arrrg.de/#ATGC>

Verändere das Programm zur Ziegelreihe so, dass es in beliebig großen Welten funktioniert.



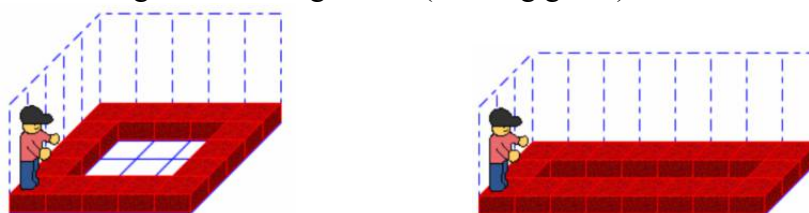
bedingte Wiederholung-b) <https://karol.arrrg.de/#GEWU>

Verändere das Programm zur **Diagonale** so, dass es in beliebig großen, quadratischen Welten funktioniert.



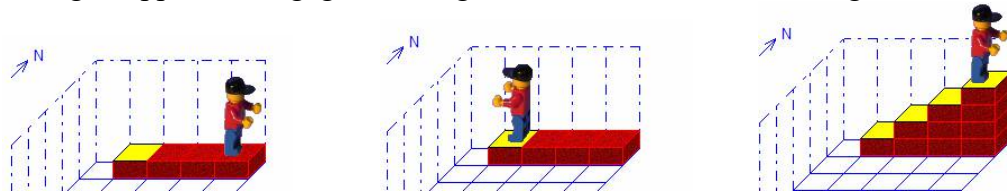
bedingte Wiederholung-c) <https://karol.arrrg.de/#XRQK>

Erzeuge ein **Becken**, das so groß, wie die gesamte (beliebig große) Welt ist.



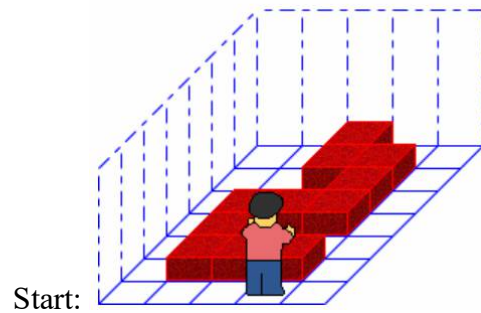
bedingte Wiederholung-Zusatz) <https://karol.arrrg.de/#25AM>

Baue eine Ziegeltrappe bis zur gegenüberliegenden Wand. Beachte die dargestellte Idee.



bedingte Wiederholung-Zusatz-2) <https://karol.arrrg.de/#47R6>

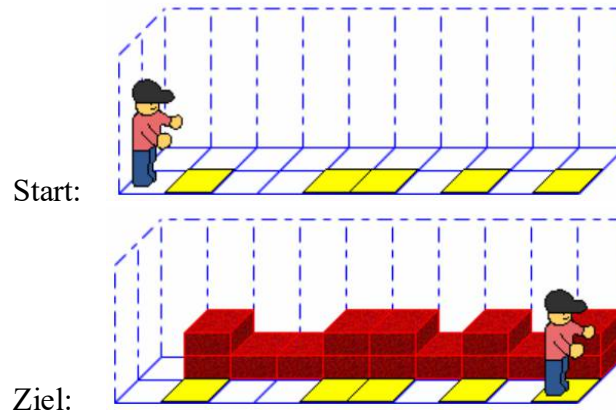
Vor Karol liegt eine Spur aus Ziegeln bis zur gegenüberliegenden Wand. Die Ziegel liegen jeweils ohne Lücke direkt hintereinander oder nebeneinander (nicht über die Diagonale). Sammle alle Ziegel ein.

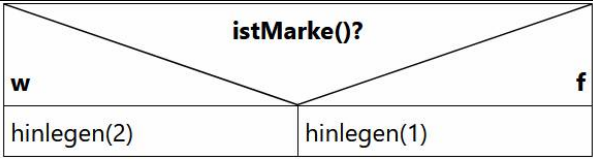
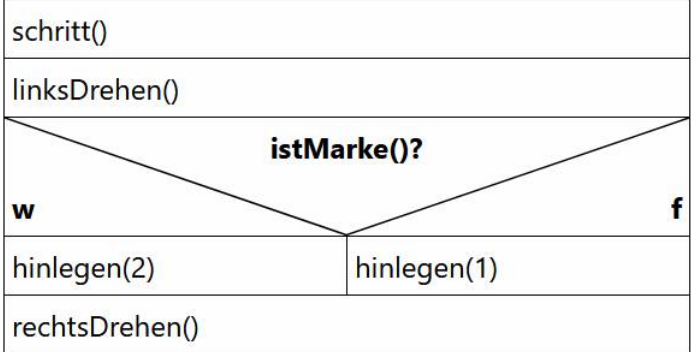


3 Verzweigungen mit if-else-Anweisung und if-Anweisung

Beispielprogramm: zweiseitige Verweigung <https://karol.arrrg.de/#3JPK>

Karol soll den Weg kennzeichnen.
 Wenn es eine Marke gibt,
 so legt er 2 Ziegel darüber ab,
 sonst
 legt er 1 Ziegel darüber ab.

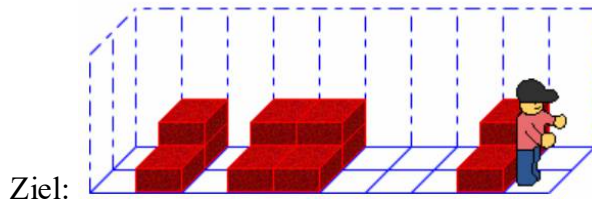
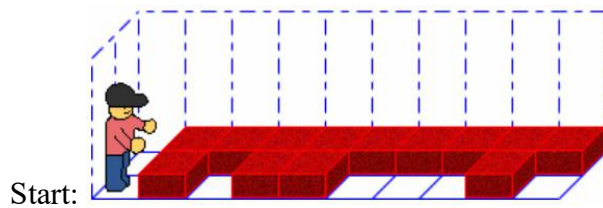


Quelltext	Struktogramm
<pre> if karol.istMarke(): karol.hinlegen(2) else: karol.hinlegen(1) </pre>	
<pre> karol = Robot() for i in range(9): karol.schritt() karol.linksDrehen() if karol.istMarke() : karol.hinlegen(2) else: karol.hinlegen(1) karol.rechtsDrehen() </pre>	<p>wiederhole 9 mal</p> 

Aufgaben

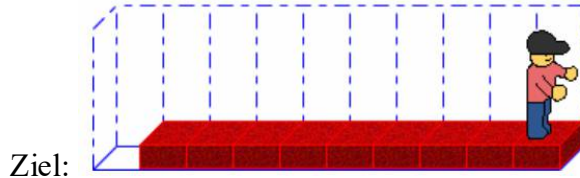
Zweiseitige Verzweigung-a) <https://karol.arrrg.de/#ZY9H>

Immer, wenn ein Ziegel da liegt, legt Karol oben einen Ziegel dazu, sonst nimmt er oben den Ziegel weg.



Einseitige Verzweigung-a) <https://karol.arrrg.de/#3ZBX>

Karol die Reihe mit Ziegeln fertig ausfüllen. Wenn kein Ziegel da ist, soll er einen Ziegel hinlegen.

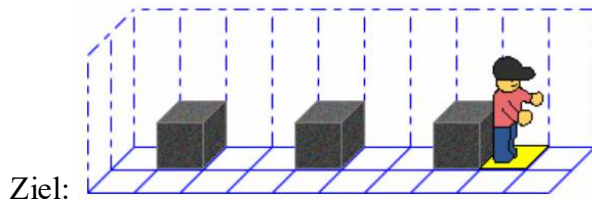
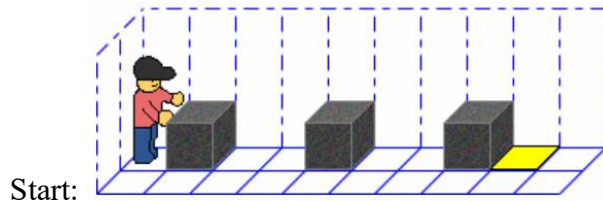


Zweiseitige Verzweigung b) <https://karol.arrrg.de/#Z4WZ>

Karol soll sein Handy (Markierung) finden und auf diesem stehen bleiben.

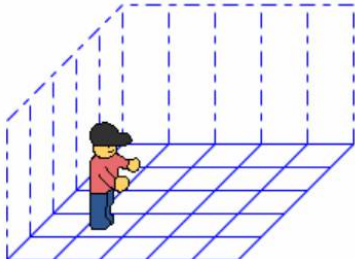
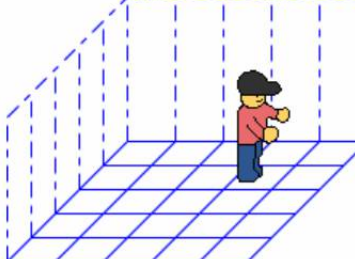
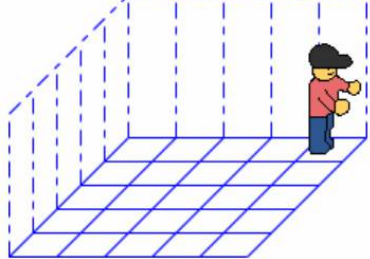
Er muss also entweder den Quader umlaufen oder einfach einen Schritt gehen.

Einen Quader kann man mit der Bedingung *IstWand* aufspüren.



4 Eigene Methode mit def

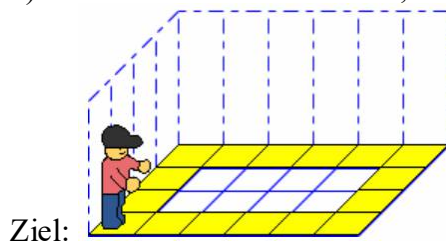
Beispielprogramm: eigene Methode <https://karol.arrrg.de/#CGYN>

Definition einer Methode	Anwendung im Programm	Anwendung in anderen Methoden
<pre>karol = Robot() def geheSchräg(): karol.schritt() karol.linksDrehen() karol.schritt() karol.rechtsDrehen()</pre>	<pre>geheSchräg() geheSchräg() geheSchräg()</pre>	<pre>def geheDiagonale(): while karol.nichtIstWand(): geheSchräg()</pre>
		

Aufgaben:

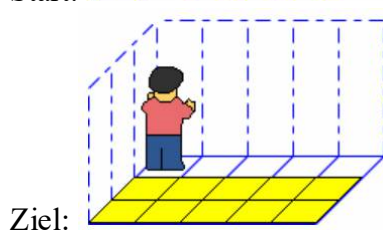
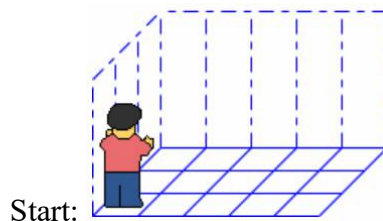
Eigene Methode-1) <https://karol.arrrg.de/#8CDS>

- Schreibe eine eigene Methode, bei der Karol eine Zeile bis ans Ende läuft und dabei mit Marken auslegt. → **markiereBisEnde()**
- Benutze die neue Methode, um eine beliebige rechteckige Welt zu **umrahmen**.



Eigene Methode-2) <https://karol.arrrg.de/#47JZ>

- Schreibe eine eigene Methode, bei der Karol eine Zeile bis ans Ende abläuft → **laufeBisEnde()**
- Benutze die neuen Methoden aus 1) und 2), um eine beliebige rechteckige Welt bis auf die oberste Zeile **mit Marken auszulegen**.



Eigene Methode-Zusatz) <https://karol.arrrg.de/#4SCC>
Entwickle eigene Methoden, um beliebige quadratische
Welten zu **umrahmen** und mit einem **Diagonalen-Kreuz**
zu **markieren**.

