

# Arbeitsheft Algorithmen



Ein kompetenzorientierter Zugang auf Basis der  
Aufgaben des Jugendwettbewerbs Informatik

**Konzeption des Arbeitsheftes:**

**Dr. Wolfgang Pfeffer**

Dominicus-von-Linprun-Gymnasium Viechtach

E-Mail: [schule@pfeffer-wolfgang.de](mailto:schule@pfeffer-wolfgang.de)



**Herausgegeben von den Bundesweiten Informatikwettbewerben (BWINF).**

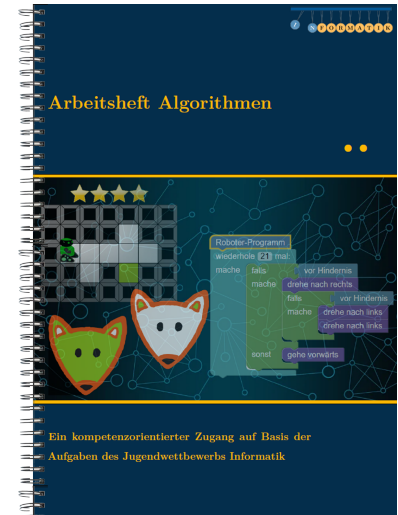
Dieses Arbeitsheft ist lizenziert unter einer **Creative Commons CC BY-SA Lizenz** (Namensnennung – Weitergabe unter gleichen Bedingungen).

Explizit ausgenommen von der CC-Lizenz sind alle enthaltenen Bilder, die keine Screenshots der Plattform [www.jwinf.de](http://www.jwinf.de) darstellen. Diese Bilder sind der Plattform [www.pixabay.com](http://www.pixabay.com) entnommen und stehen unter der dort aufgeführten Lizenz.



## I. Über das Arbeitsheft

In den **Lehrplänen der Unterstufe** im Bereich **Informatik** resp. **Natur und Technik** nimmt das Thema „**Algorithmik**“ in vielen Bundesländern eine immer größer werdende Rolle ein. Besonderer Beliebtheit erfreuen sich immer mehr „bausteinorientierte Programmiersysteme“ wie **Scratch**, **Snap**, **Blockly**, etc. Als ideale Ergänzung zum regulären Unterricht und zur weiteren Förderung von Informatik-Talenten erweist sich jedes Jahr der **Jugendwettbewerb Informatik**. Auf der Seite [www.jwinf.de](http://www.jwinf.de) finden sich viele Trainingsaufgaben für den Wettbewerb, wobei diese eigentlich nahezu perfekt mit den Inhalten der Lehrpläne übereinstimmen. Intention des Arbeitsheftes ist es, interessierten Lehrkräften die Möglichkeit zu bieten, die Unterrichtsinhalte direkt anhand der Aufgaben auf der Plattform [www.jwinf.de](http://www.jwinf.de) umzusetzen.



Gemäß der bisher gemachten Erfahrungen im Unterricht, führt die Behandlung der Inhalte mithilfe von Programmen mit bausteinorientierten Programmiersprachen auf Schülerseite einerseits zu großer Motivation, problematisch ist andererseits allerdings die schriftliche Dokumentation der Programme. Das Arbeitsheft versucht genau dieser Problematik entgegenzuwirken und bietet zu den ausgewählten Aufgaben eine Auflistung der verschiedenen Bausteine sowie ausreichend Platz, das erstellte Programm (z.B. als Struktogramm) auch im Heft schriftlich festzuhalten. Die Aufgaben sind zudem flankiert von Merke-Kästen, in denen die wichtigsten Begriffe bzw. Bausteine erklärt und visualisiert sind. Aktuell umfasst das Arbeitsheft folgende Teilbereiche:

- (a) Erste Schritte: Anweisung, Sequenz und Programm
- (b) Wiederholung mit fester Anzahl
- (c) Bedingte Anweisung
- (d) Wiederholung mit Bedingung
- (e) Zweiseitig bedingte Anweisung
- (f) Vermischte und schwere Aufgaben

Die im Arbeitsheft dokumentierten Aufgaben finden sich auf der Seite [www.jwinf.de](http://www.jwinf.de) unter Trainingsaufgaben. Eine Lehrkraft kann über den **Lehrkraft-Login** einen **Gruppencode** für seine / ihre Klassen erstellen und kann sich hierdurch die Ergebnisse der Schülerinnen und Schüler zu den jeweiligen Aufgaben anzeigen lassen. Die Schüler bekommen durch erstmaligen Login mit dem Gruppencode einen individuellen Login-Code, mit dem sie sich fortan anmelden können. Die bereits erfolgreich absolvierten Aufgaben bleiben somit für die Schülerinnen und Schüler bei erneuter Anmeldung sichtbar.

Falls Sie Fragen oder Anregungen zum Arbeitsheft haben, richten Sie diese gerne an den Verfasser.



## II. Algorithmik – Bausteinorientierte Programmiersprachen

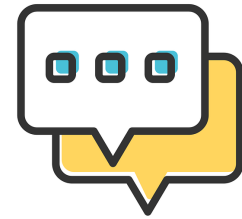
### Kompetenzerwartungen:

- Analyse und Strukturierung von Problemstellungen
- Entwicklung von Algorithmen unter Verwendung von Kontrollstrukturen
- Nutzung der algorithmischen Bausteine zur Lösung von Problemstellungen mithilfe eines geeigneten Programmierwerkzeuges



Auch in unserem Alltag finden wir fast überall verschiedene Folgen von Anweisungen, sei es beim Rezept zum Kuchen backen, beim Montieren von Schränken, beim Basteln oder beim Wäsche waschen. Diese Anweisungen beschreiben uns Schritt für Schritt den genauen Ablauf von bestimmten Tätigkeiten und geben uns eine Reihenfolge vor, in der wir einzelne Aufgaben erledigen müssen.

All diese Anleitungen führen wir aus, um am Ende ein gewünschtes Ergebnis zu erhalten. Solche Folgen von Anweisungen nennt man **Algorithmen**.

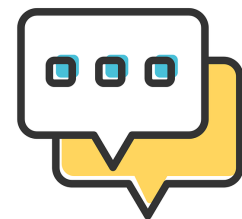


### Merke (Algorithmus)

Ein **Algorithmus** ist eine **endliche** Folge von **eindeutigen** und **ausführbaren** Anweisungen zur Lösung eines Problems.

Unter gleichen Voraussetzungen liefert die Ausführung eines Algorithmus stets das gleiche Ergebnis.

Ein Teilgebiet der Informatik beschäftigt sich mit dem **Programmieren**. Letztendlich liegt jedem Programm, das ein Programmierer erstellt, ein **Algorithmus** zugrunde. Es gilt hierbei ein Problem mithilfe von Programmierbausteinen durch logisches Denken zu lösen. In diesem Abschnitt lernst du mithilfe einer bausteinorientierten Programmiersprache, selbstständig Programme zu entwerfen, die gegebene Problemstellungen lösen. Wir nutzen hierbei Trainingsaufgaben der Seite [www.jwinf.de](http://www.jwinf.de) (Jugendwettbewerb Informatik).



### Info (Die Seite [www.jwinf.de](http://www.jwinf.de))

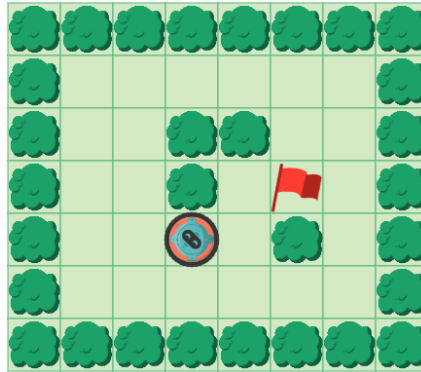
Die Seite des **Jugendwettbewerbes Informatik** bietet viele Aufgaben, die sich für unseren Unterricht bestens eignen. Du erhältst einen Gruppencode und kannst dich damit anmelden. Notiere dir hier anschließend den dir zugewiesenen **Login-Code**, mit dem du dich ab jetzt immer auf dieser Seite anmelden kannst:

**Arbeitsauftrag 1: Erste Programme**


Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu **Arbeitsheft Algorithmen**.

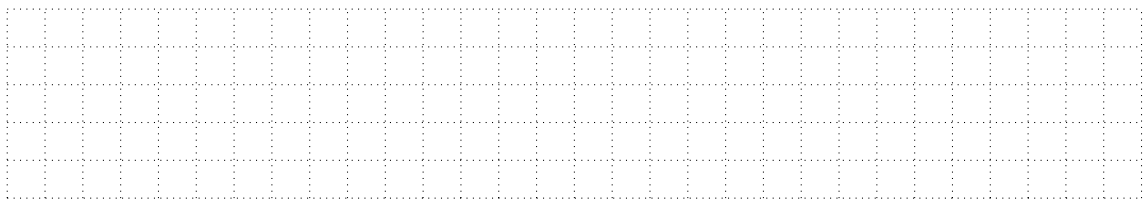
(a) Bausteine zusammensetzen

Programmiere den Roboter: Der Roboter soll die rote Fahne erreichen, ohne gegen die Hindernisse zu laufen:



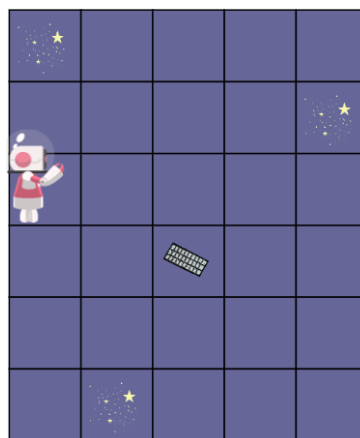
Dir stehen hierfür die Bausteine **gehe nach rechts** und **gehe nach oben** zur Verfügung.

Klicke anschließend auf das -Symbol, um das Programm überprüfen zu lassen.

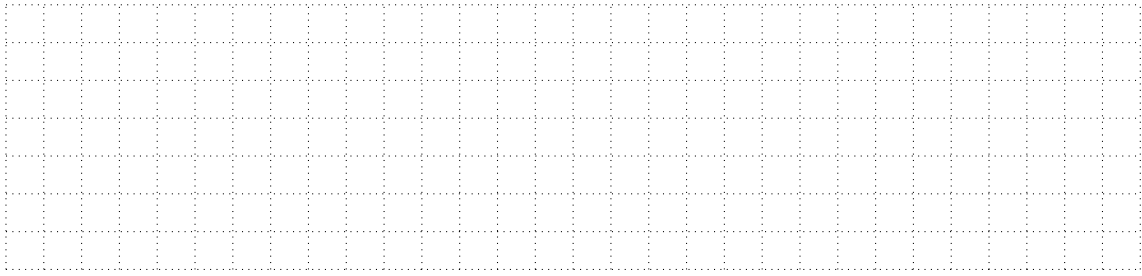


(b) Befehlsfolgen - Objekte im Weltraum

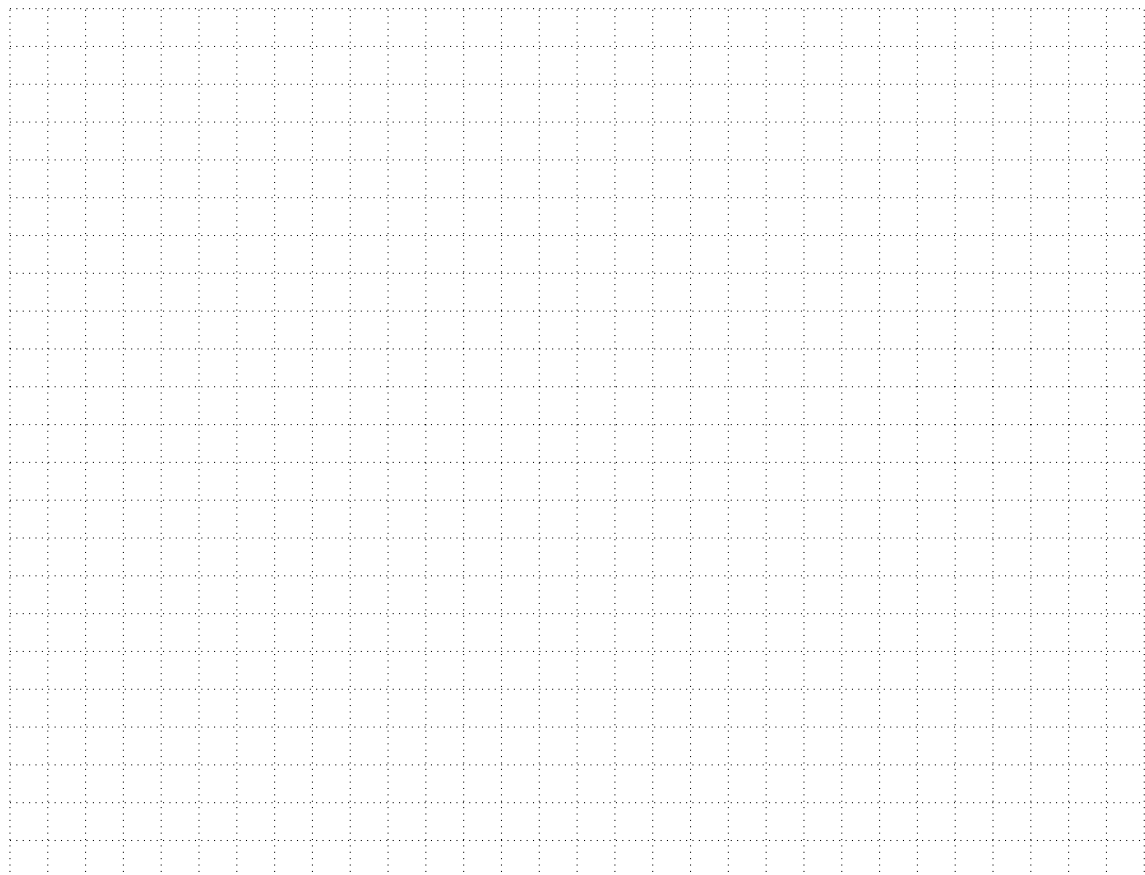
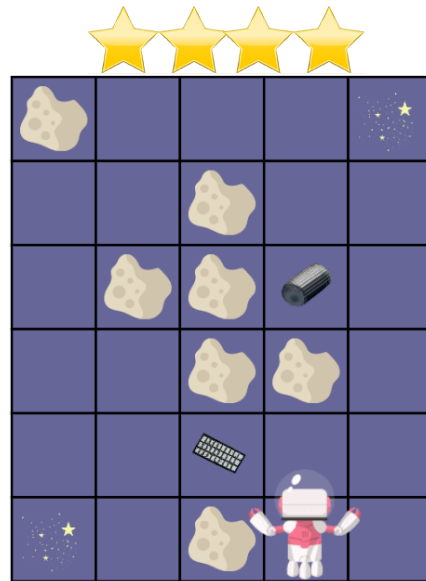
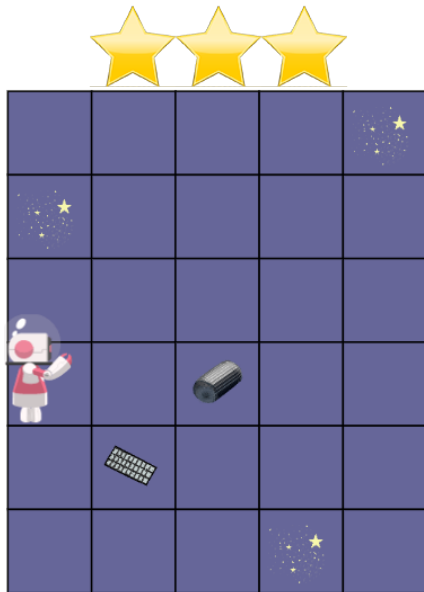
Programmiere den Roboter. Sammle das im Weltraum verlorene Objekt ein. Nutze den Baustein **hebe Objekt auf**, um das Objekt einzusammeln.



Dir stehen hierfür die Bausteine **drehe nach links**, **drehe nach rechts**, **gehe vorwärts** und **hebe Objekt auf** zur Verfügung.

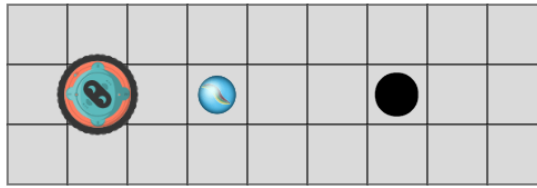


Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:



(c) Befehlsfolgen - Murmeln

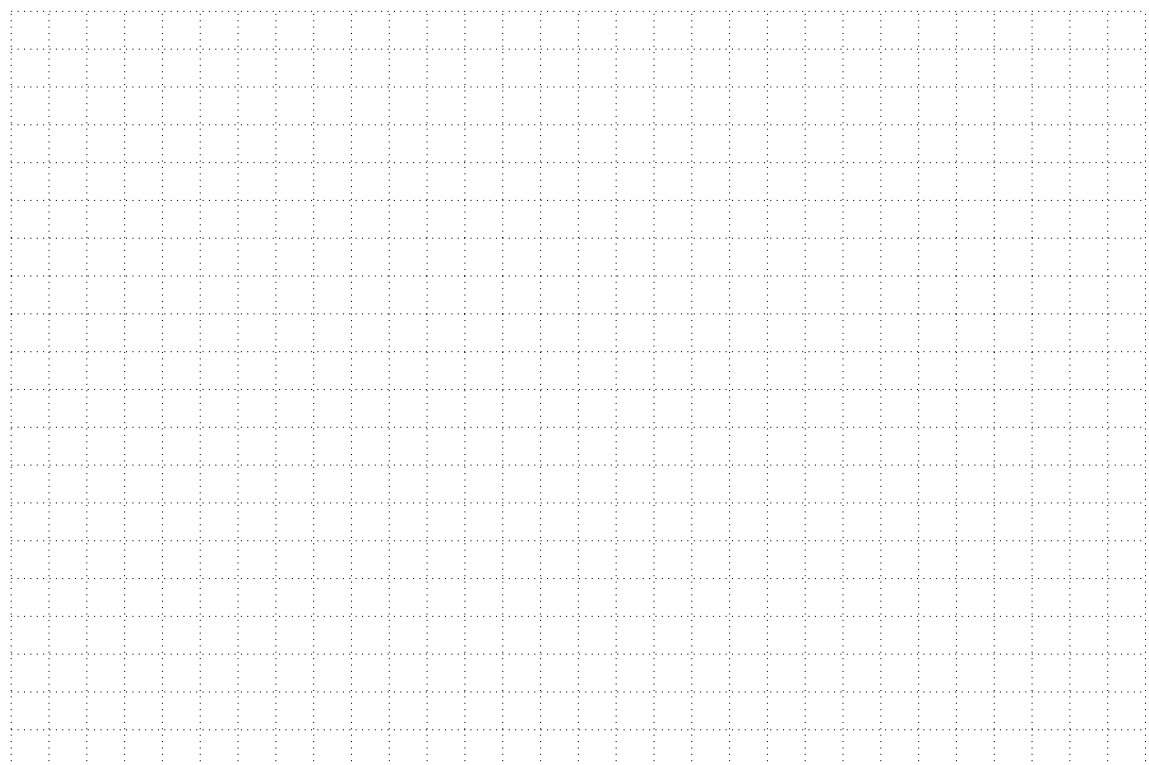
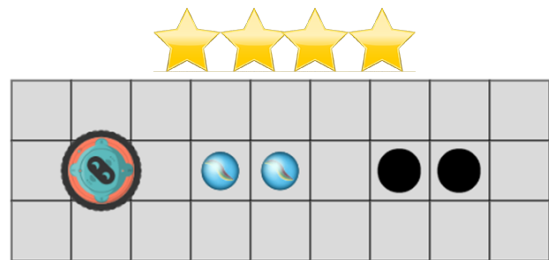
Programmiere den Roboter: Er soll die Murmel im Loch ablegen:



Dir stehen hierfür die Bausteine **gehe nach rechts** , **gehe nach links** , **hebe Murmel auf** und **lege Murmel ab** zur Verfügung.



Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:

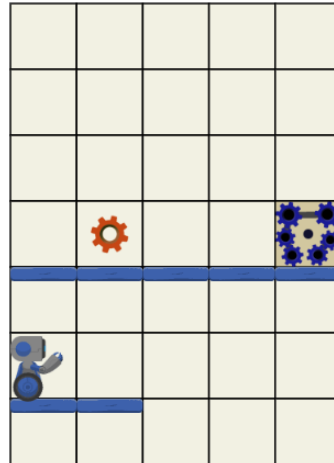


**Arbeitsauftrag 2: Plattformen und Lichter anzünden**

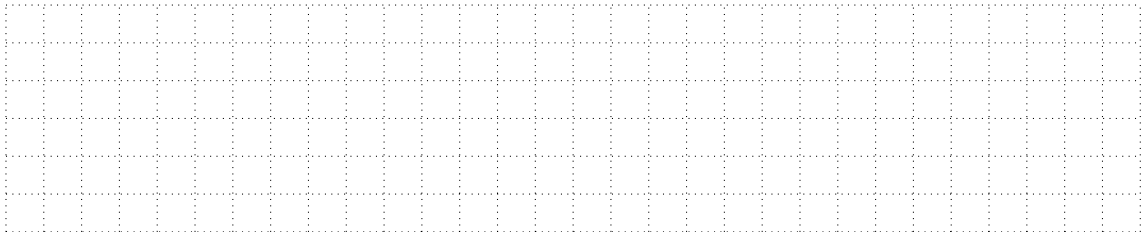
Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu **Arbeitsheft Algorithmen**.

(a) Plattformen

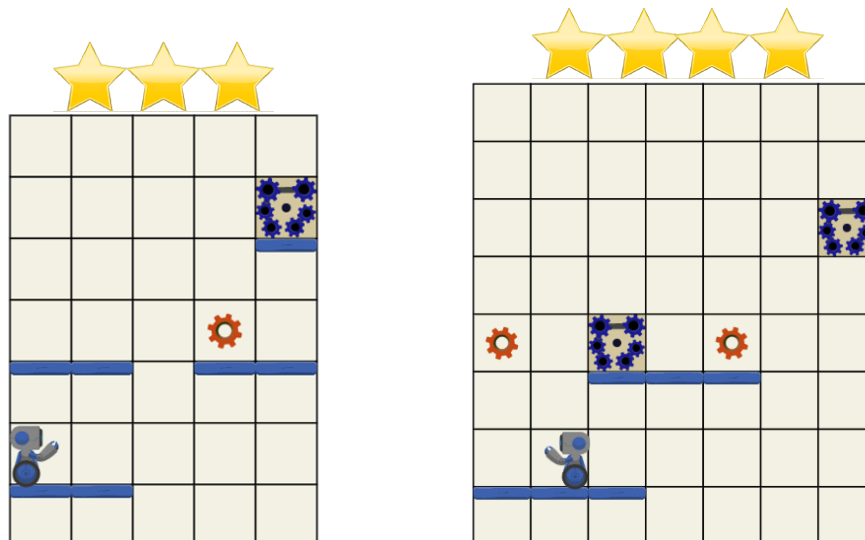
Programmiere den Roboter. Der Roboter soll alle Zahnräder in die Maschine einsetzen:



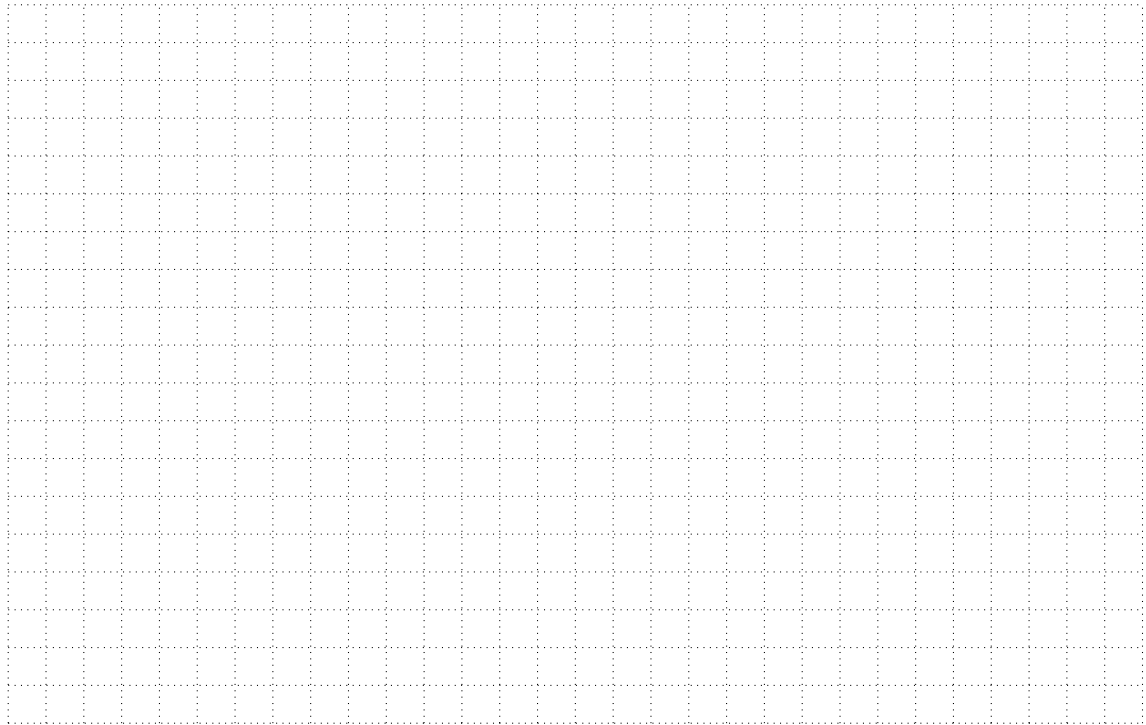
Dir stehen hierfür die Bausteine **gehe vorwärts**, **hebe Zahnrad auf**, **baue Zahnrad ein** und **spring hoch** zur Verfügung.



Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Bei der 3-Sterne-Variante steht dir zusätzlich der Baustein **baue eine Plattform vorne** und bei dir 4-Sterne Variante nochmal zusätzlich die Bausteine **drehe um** sowie **baue eine Plattform oben** zur Verfügung. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:

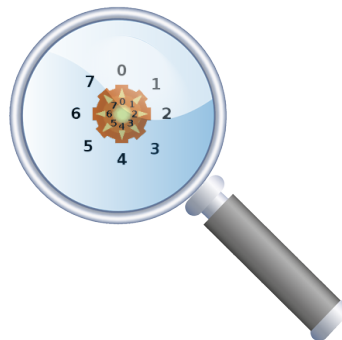
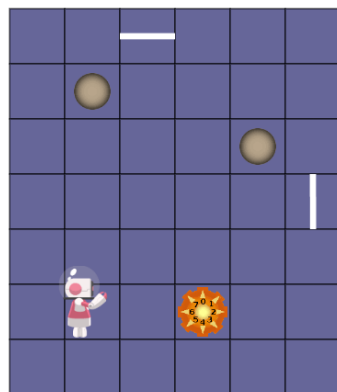




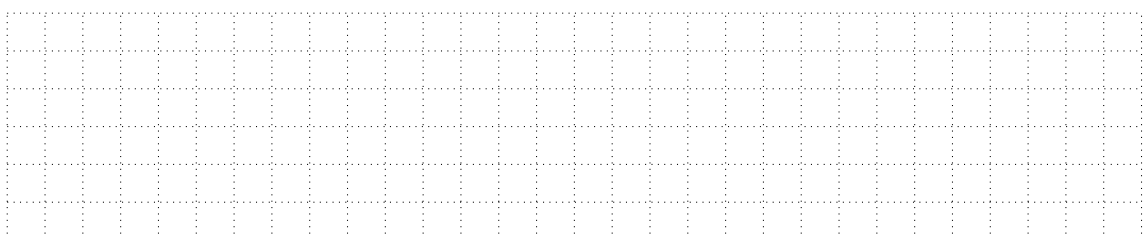


(b) **Lichter anzünden**

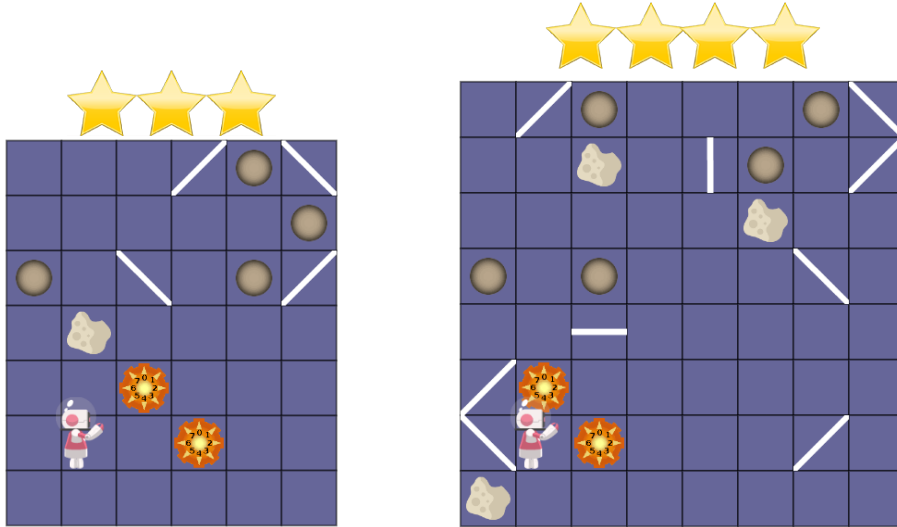
Programmiere den Roboter. Alle Punkte müssen mit dem Laserstrahl angeschaltet werden. Laser müssen dabei über die Spiegel (dargestellt als weiße Balken) gelenkt werden. Ein Laserstrahl beleuchtet alle Stellen auf seinem Weg. Einmal erleuchtete Stellen bleiben erleuchtet. Um einen Laserwerfer zu verwenden, muss der Roboter auf dem gleichen Feld stehen. Die Richtung wird durch eine ganze Zahl zwischen 0 und 7 vorgegeben. Die Richtungen werden auf dem Laserwerfer angezeigt:



Dir stehen hierfür die Bausteine `gehe vorwärts`, `richte Laserstrahl aus nach` und `Ziffer` zur Verfügung.



Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Bei diesen Versionen stehen dir zusätzlich die Bausteine **drehe nach links** und **drehe nach rechts** zur Verfügung. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:



A large grid of dotted lines for writing the solution code.

**Merke (Anweisung, Sequenz und Programm)**

Der Roboter wird durch **Anweisungen** gesteuert, die man als Methodenaufrufe betrachten kann.

Eine **Sequenz** ist eine Folge von Anweisungen, bei deren **Ablauf** diese in der angegebenen Reihenfolge abgearbeitet werden.

Ein **Programm** ist eine Folge von Anweisungen in einer **Programmiersprache**, die von einem Gerät ausgeführt werden können.

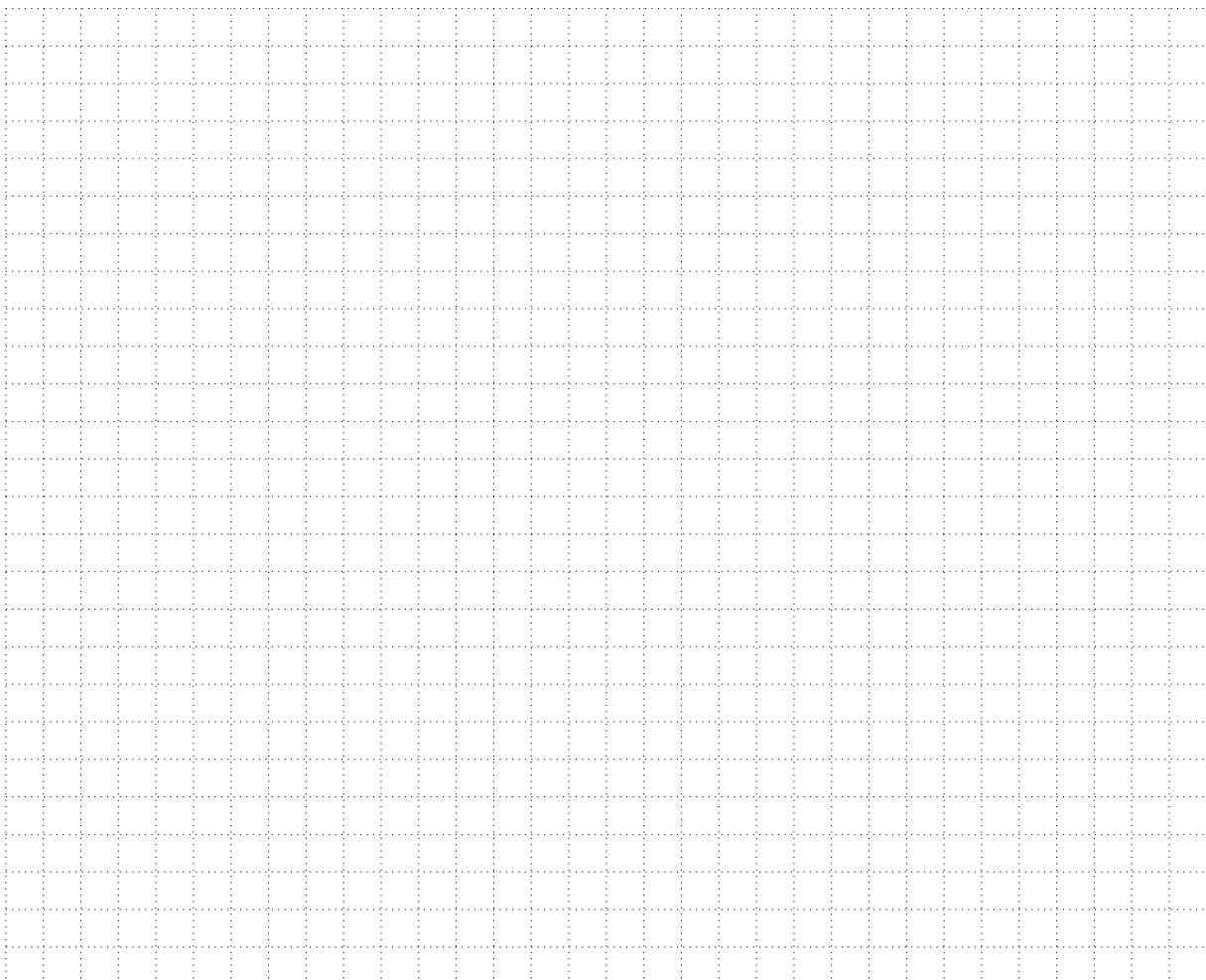
**Merke (Struktogramme)**

Zur graphischen Darstellung eines Programmablaufs verwendet man sogenannte **Struktogramme**. Damit lassen sich Programmabläufe unabhängig von konkreten Programmiersprachen übersichtlich darstellen.

Für eine **Sequenz** sieht das Struktogramm folgendermaßen aus.

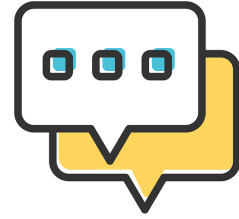
Anweisung 1
Anweisung 2
Anweisung 3
...

*Platz für Notizen:*



## 2.1 Wiederholung mit fester Anzahl

Im vorherigen Programm haben wir gesehen, dass unser Algorithmus mit so vielen Anweisungen sehr lang und unübersichtlich wird. Wir haben außerdem immer wieder die gleichen Anweisungen verwendet. Einfacher wäre es doch, wenn wir unserem Programm sagen könnten, dass es eine ganze Sequenz einfach z.B. 10 mal wiederholen soll. Dazu gibt es in der Algorithmik die Kontrollstruktur **Wiederholung mit fester Anzahl**. Wir können so ganze Sequenzen wiederholen, ohne sie jedes mal wieder neu schreiben zu müssen.



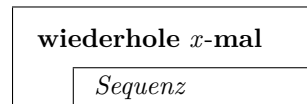
### Merke (Wiederholung mit fester Anzahl)

Die Kontrollstruktur **Wiederholung mit fester Anzahl** dient dazu, mehrfache Wiederholungen von Anweisungen oder ganzen Sequenzen einfach zu realisieren. Die Anzahl der Wiederholungen steht dabei von Anfang an bereits fest.



Wiederholung mit fester Anzahl

### Struktogramm:

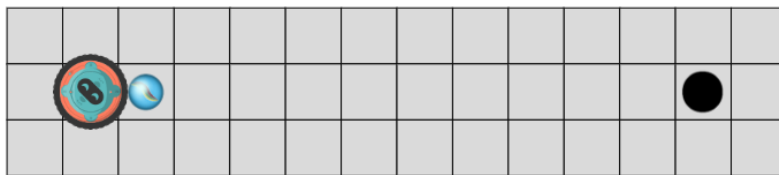


### Arbeitsauftrag 3: Wiederholung mit fester Anzahl

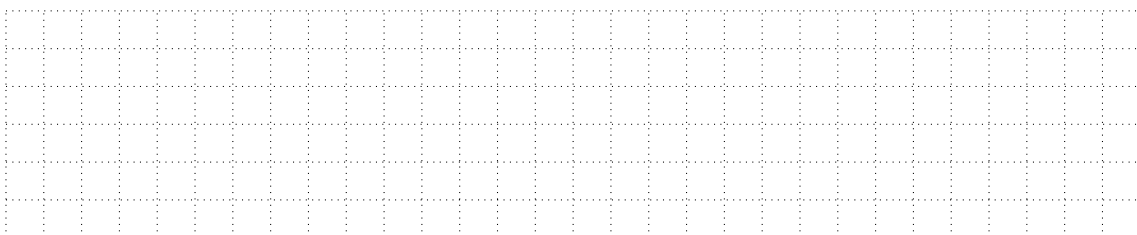
Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu Arbeitsheft Algorithmien.

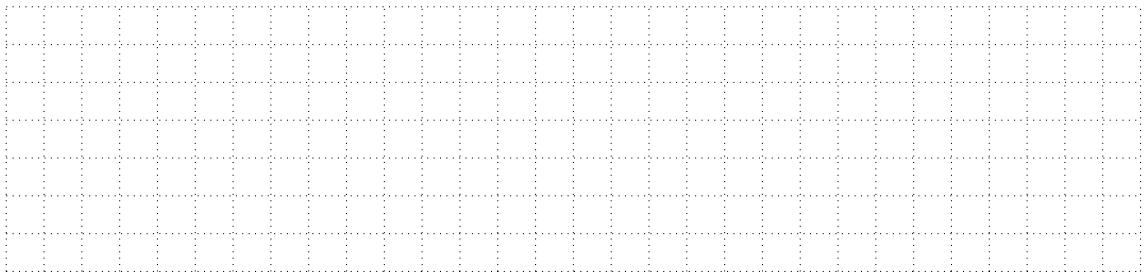
#### (a) Wiederholungen

Programmiere den Roboter: Er soll die Murmel im das Loch legen.



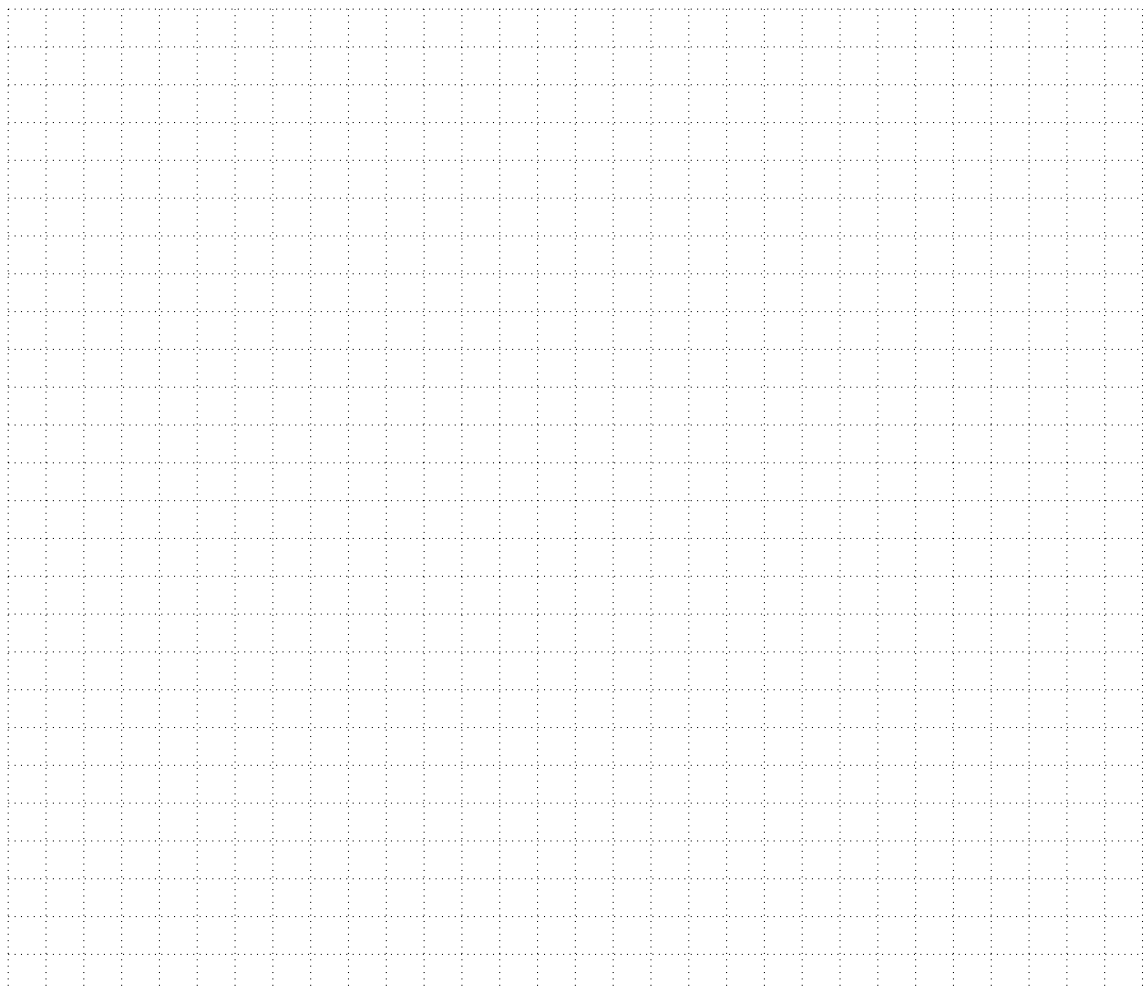
Dir stehen hierfür die Bausteine **gehe nach rechts** , **gehe nach links** , **gehe nach oben** , **gehe nach unten** , **hebe Murmel auf** , **lege Murmel ab** und **wiederhole x mal** zur Verfügung.





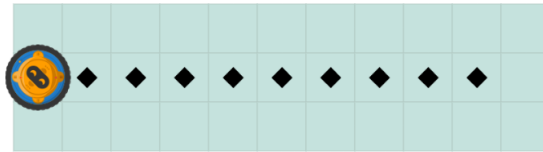
Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:

The image shows two grid-based puzzles. The left puzzle is labeled '3-Sterne' and consists of a 3x10 grid. The right puzzle is labeled '4-Sterne' and consists of a 4x10 grid. Both puzzles have a blue ball in the second row, first column. The '3-Sterne' puzzle has a red circle with a white 'B' in the second row, fifth column and a black dot in the second row, eighth column. The '4-Sterne' puzzle has a red circle with a white 'B' in the fourth row, first column and a black dot in the fourth row, eighth column. Three yellow stars are above the left puzzle, and four yellow stars are above the right puzzle.

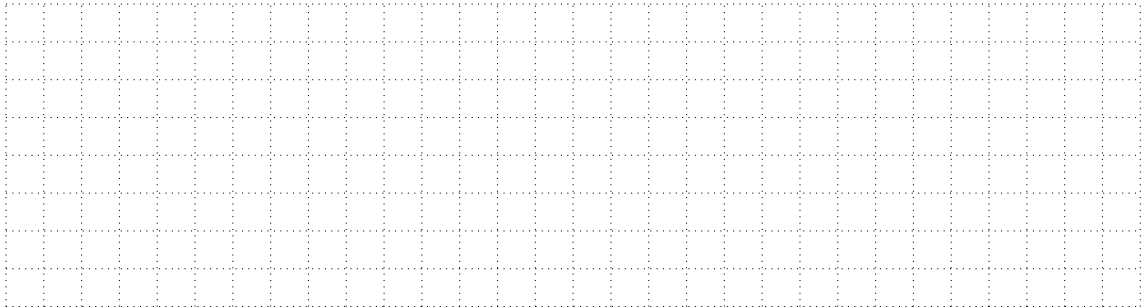


(b) Muster malen

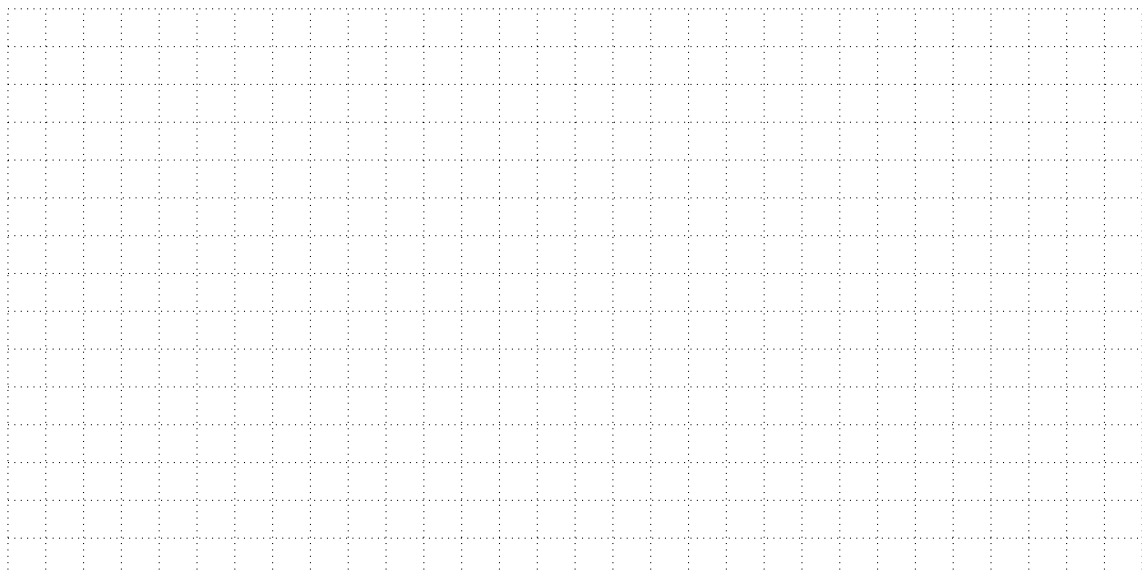
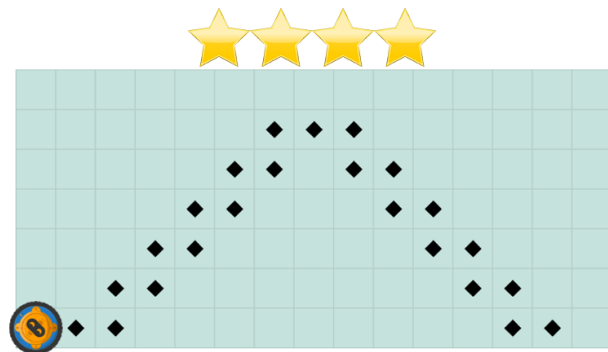
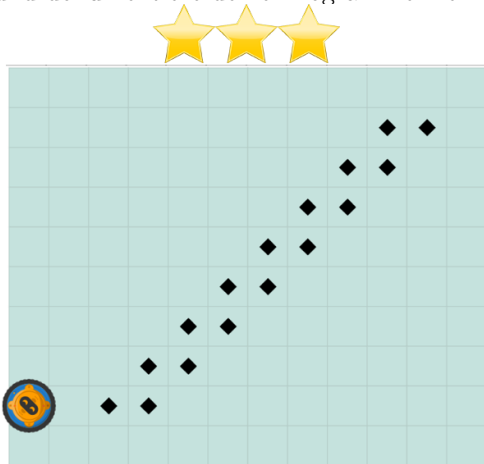
Programmiere den Roboter: Der Roboter soll die markierten Felder färben:

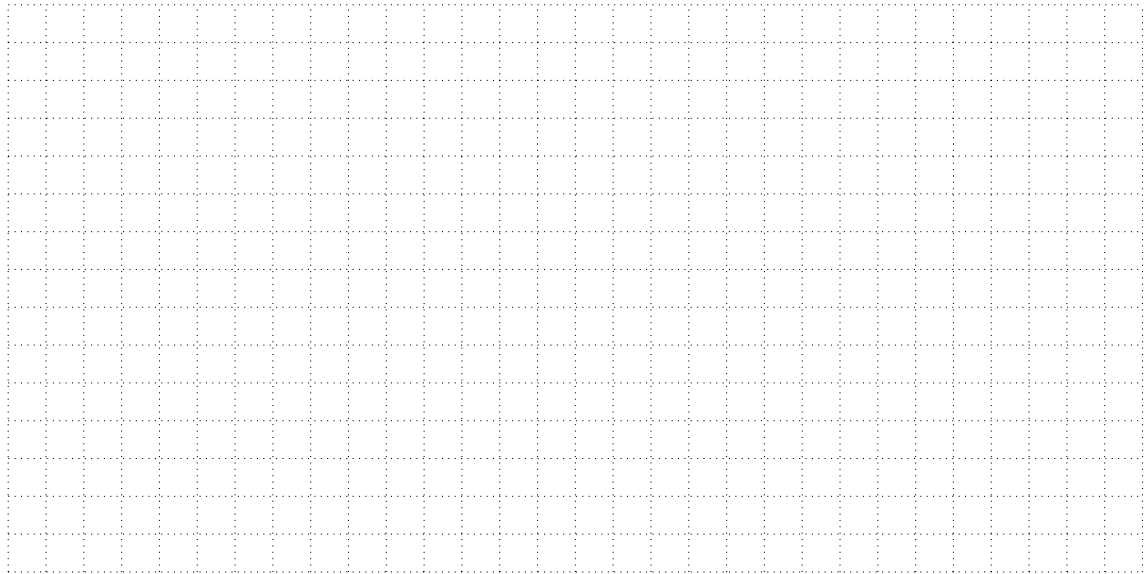


Dir stehen hierfür die Bausteine `gehe nach rechts`, `gehe nach links`, `gehe nach oben`, `gehe nach unten`, `färbe das Feld` und `wiederhole x mal` zur Verfügung.



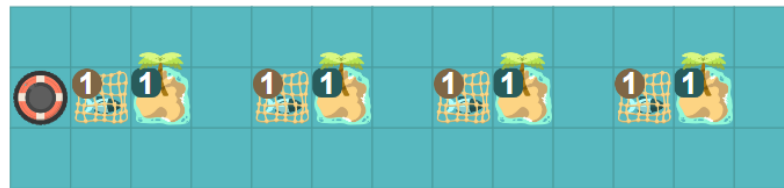
Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:



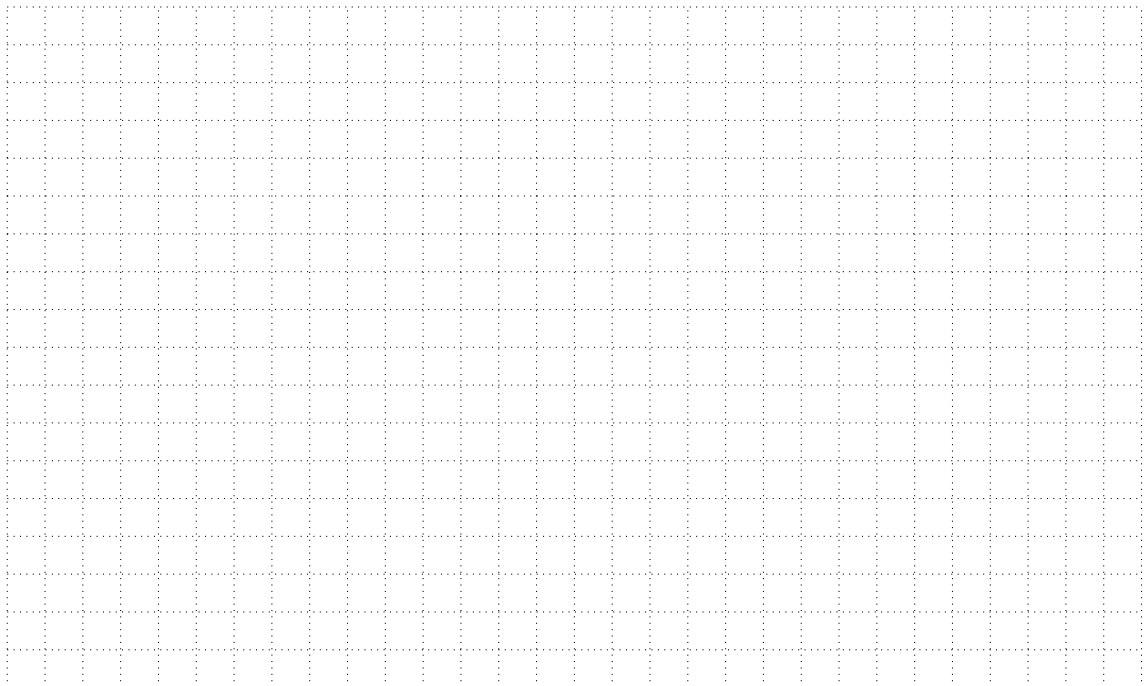


(c) Mehr Wiederholungen

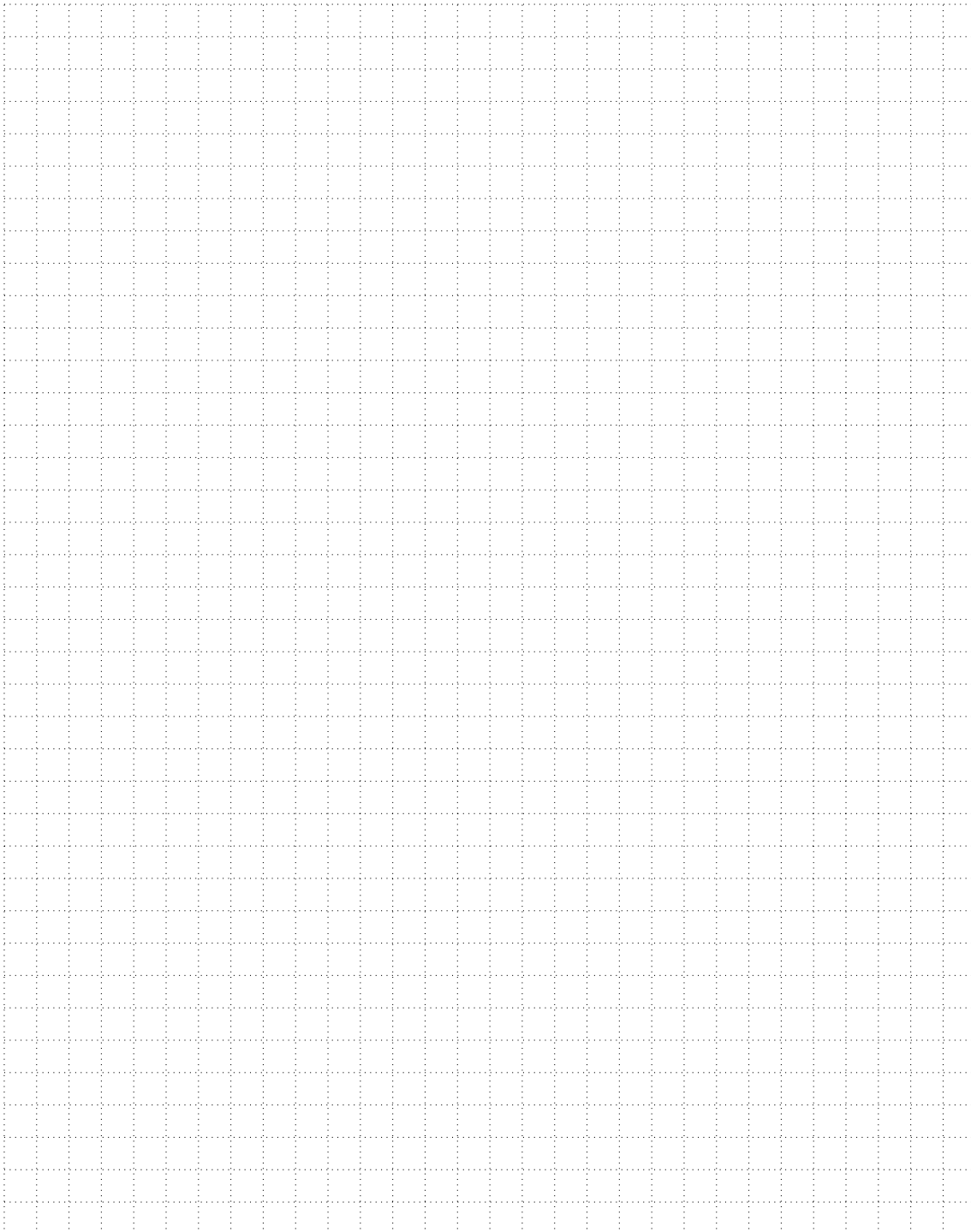
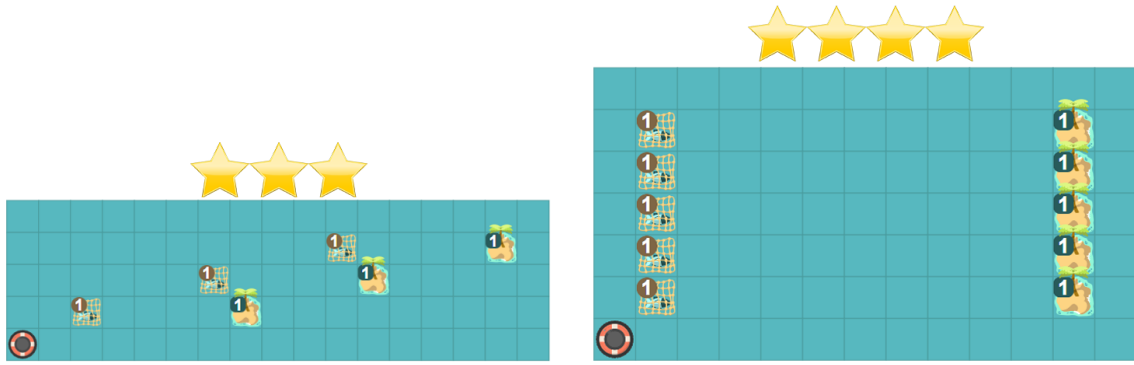
Programmiere den Roboter: Der Roboter soll jeden Fisch aufheben und auf der entsprechenden Insel ablegen. Der Roboter kann höchstens einen Fisch auf einmal tragen:



Dir stehen hierfür die Bausteine `gehe nach rechts` , `gehe nach links` , `gehe nach oben` , `fange Fisch` , `liefere Fisch ab` und `wiederhole x mal` zur Verfügung.



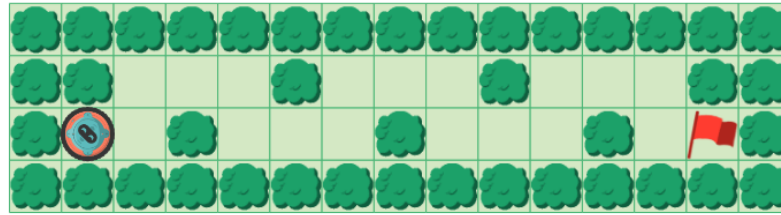
Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:



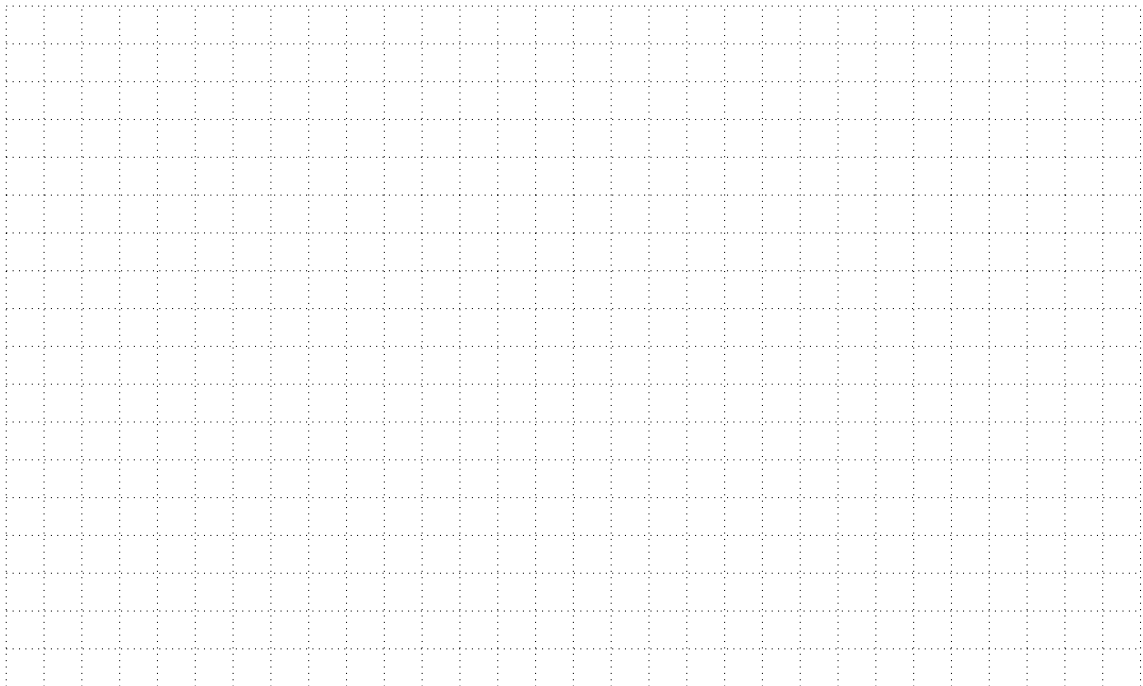


(d) Zick-Zack

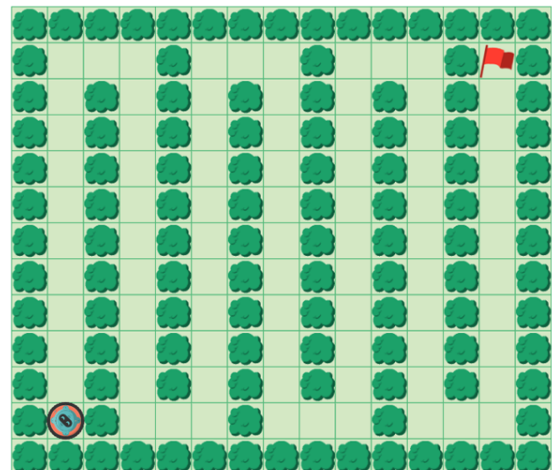
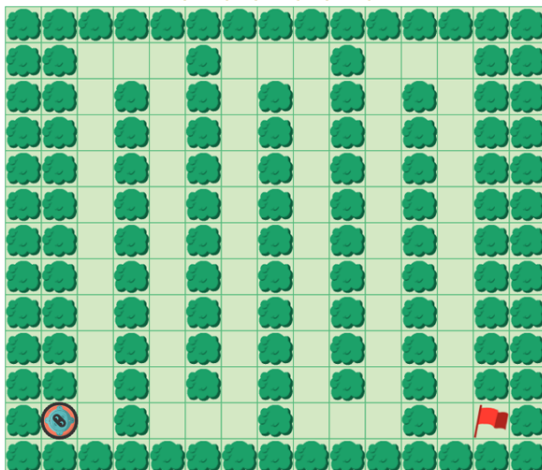
Programmiere den Roboter: Der Roboter soll die rote Fahne am Ende des Parcours erreichen, ohne gegen Hindernisse zu laufen:



Dir stehen hierfür die Bausteine `gehe nach rechts` , `gehe nach oben` , `gehe nach unten` und `wiederhole x mal` zur Verfügung.



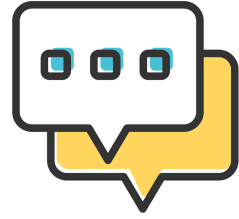
Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:





## 2.2 Bedingte Anweisung

Bisher sind wir immer davon ausgegangen, dass Sequenzen beliebig oft ausgeführt werden und erst bei Eintreten einer Bedingung beendet werden. Oftmals ist es aber gewünscht, dass Anweisungen oder Sequenzen nur dann ein einziges mal ausgeführt werden, wenn zuvor ein bestimmtes Ereignis eingetreten ist. Dies lässt sich mit der „bedingten Anweisung“ realisieren.



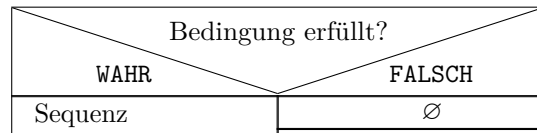
### Merke (Bedingte Anweisung)

Die Kontrollstruktur **bedingte Anweisung** dient dazu, eine Sequenz abhängig von einer Bedingung auszuführen.



*Bedingte Anweisung*

**Struktogramm:**

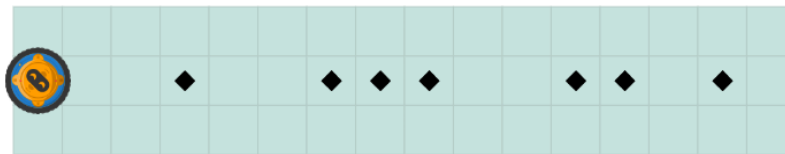


### Arbeitsauftrag 4: Bedingte Anweisung

Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu Arbeitsheft Algorithmen.

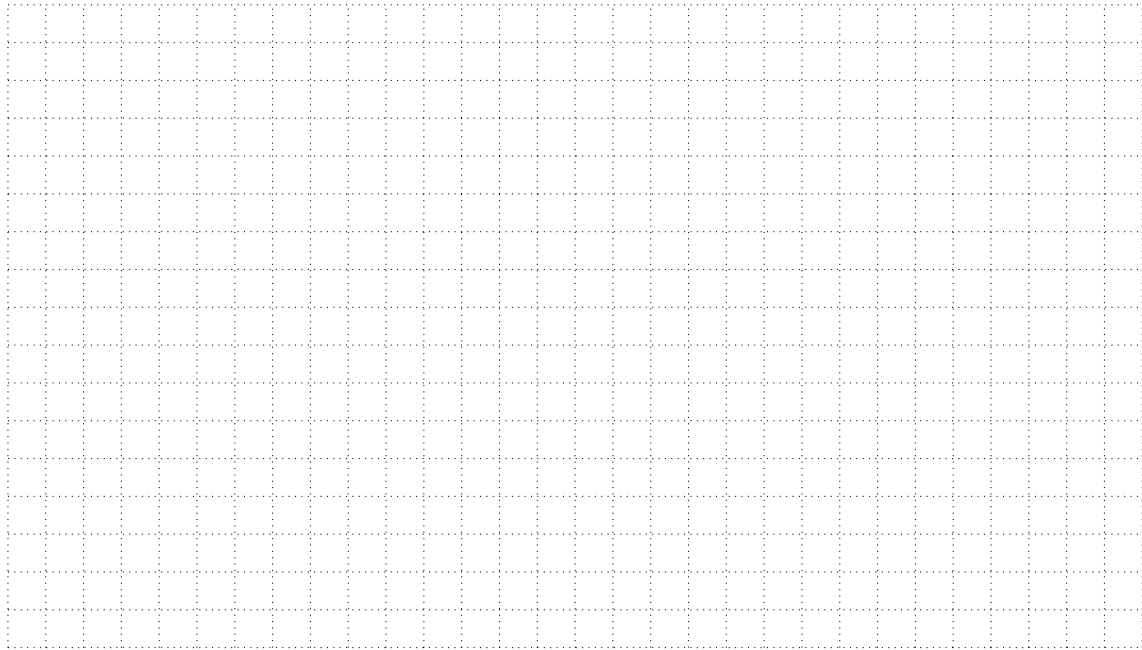
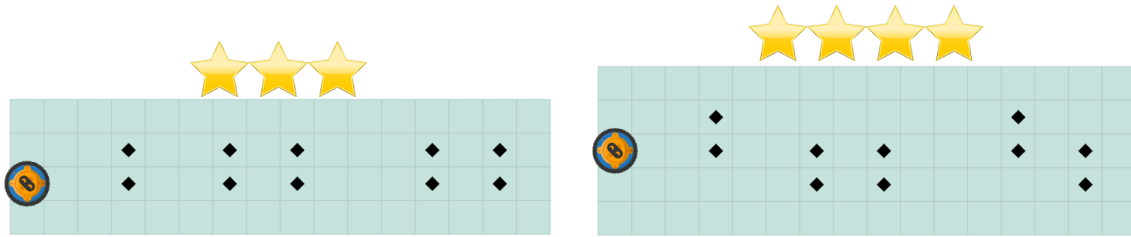
(a) Bedingte Anweisung

Programmiere den Roboter: Der Roboter soll alle markieren Felder färben:



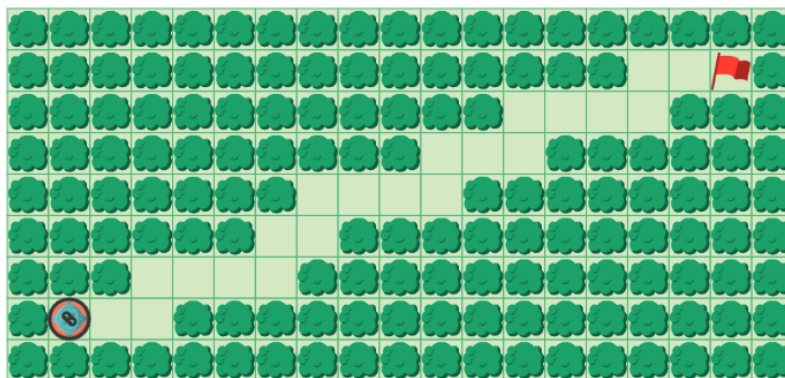
Dir stehen hierfür die Bausteine **gehe nach rechts**, **gehe nach links**, **gehe nach oben**, **gehe nach unten**, **färbe das feld**, **auf markiertem Feld**, **wiederhole x mal** und **falls x mache y** zur Verfügung.

Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:

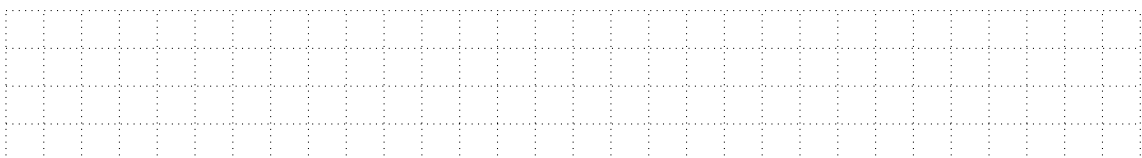


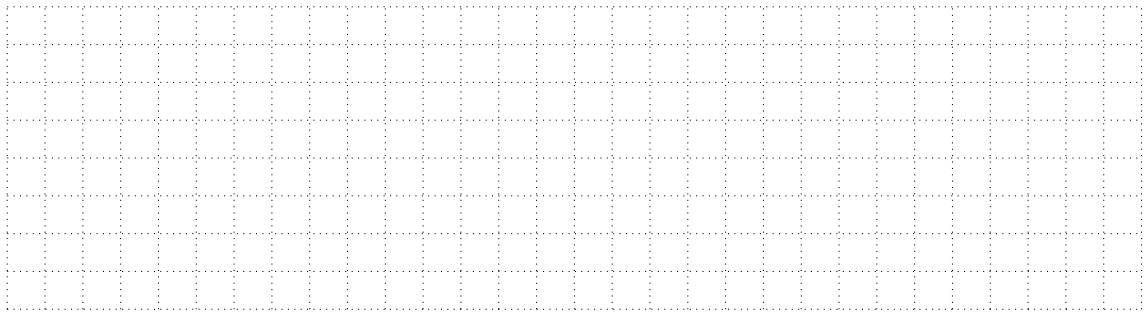
(b) Labyrinth

Programmiere den Roboter: Der Roboter soll die rote Fahne erreichen, ohne gegen die Hindernisse zu laufen:

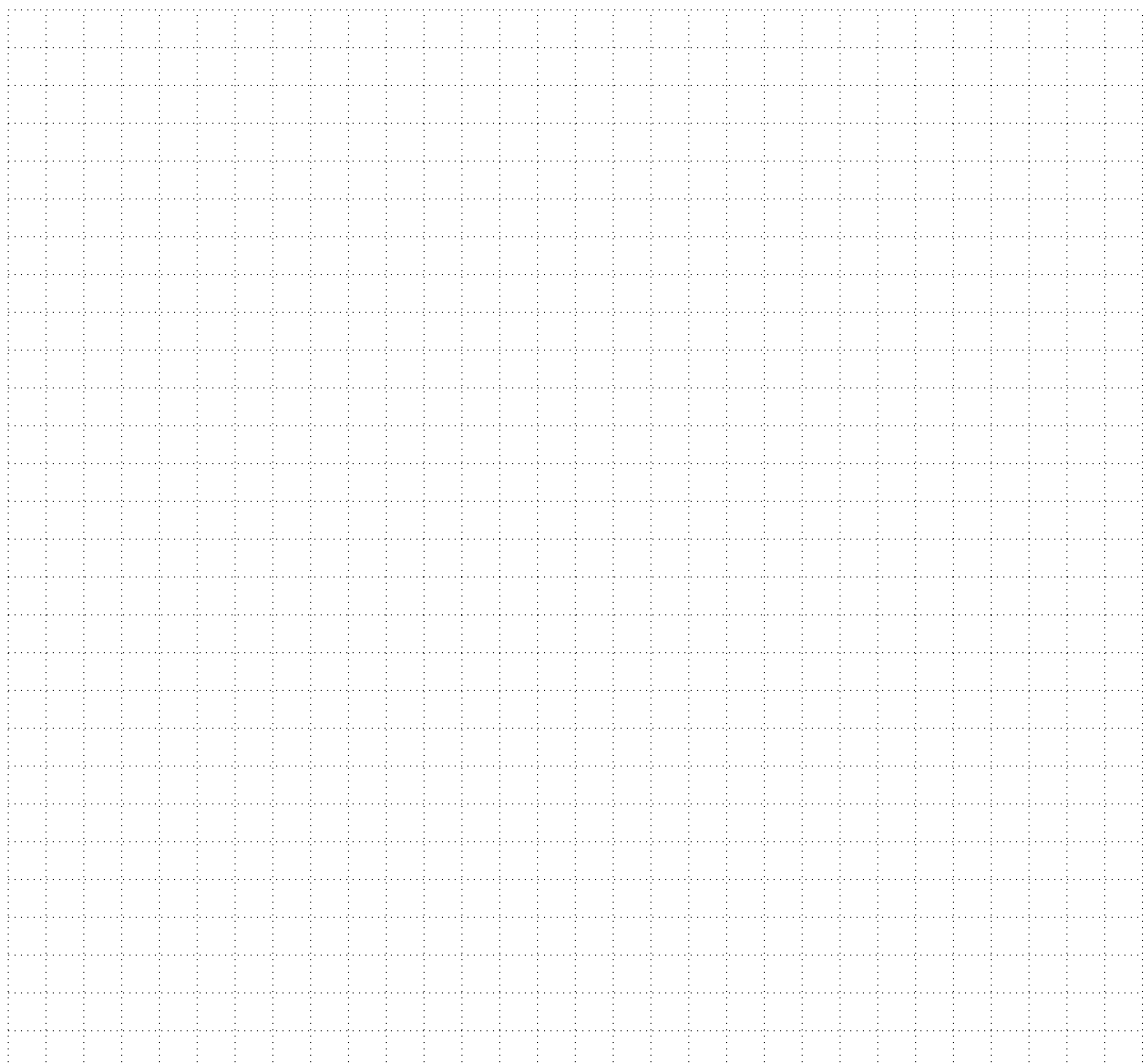
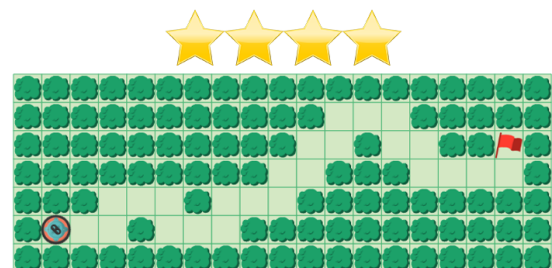
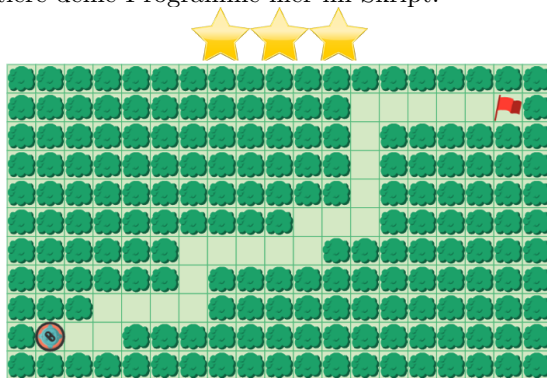


Dir stehen hierfür die Bausteine `gehe nach rechts` , `gehe nach oben` , `gehe nach unten` , `Hindernis rechts` , `wiederhole x mal` und `falls x mache y` zur Verfügung.



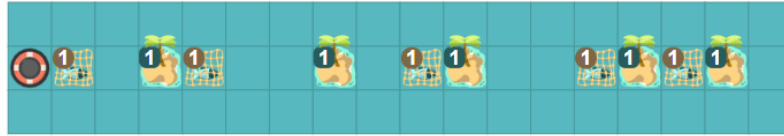


Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Hier steht dir zusätzlich der Baustein `falls x mache y sonst z` zur Verfügung. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:

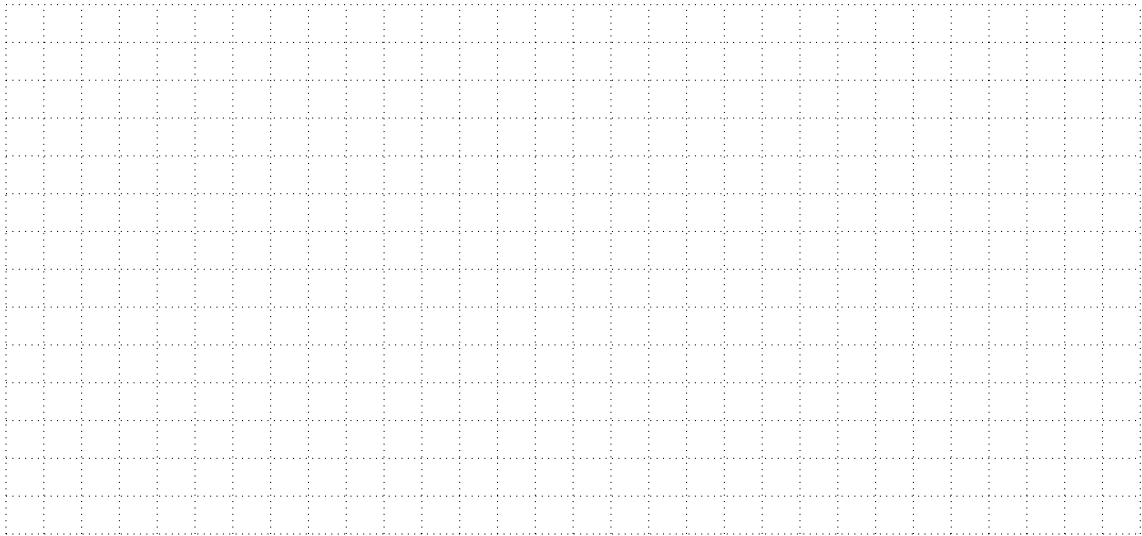


(c) Fischen

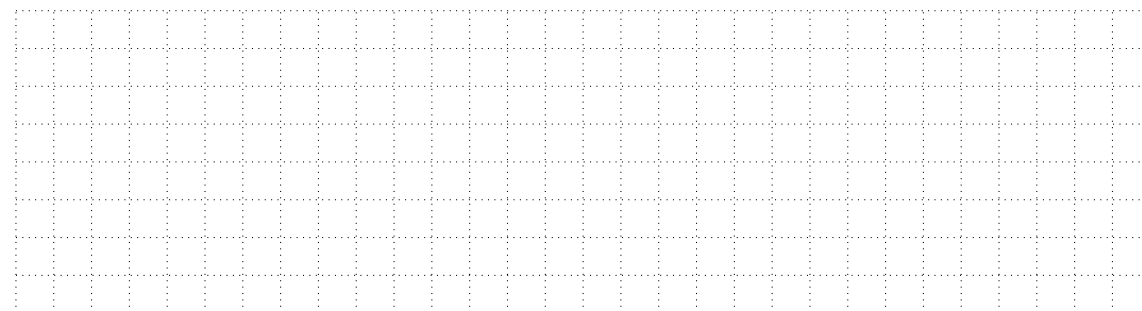
Programmiere den Roboter: Der Roboter soll jeden Fisch aufheben und auf einer Insel ablegen. Der Roboter kann höchstens einen Fisch auf einmal tragen:

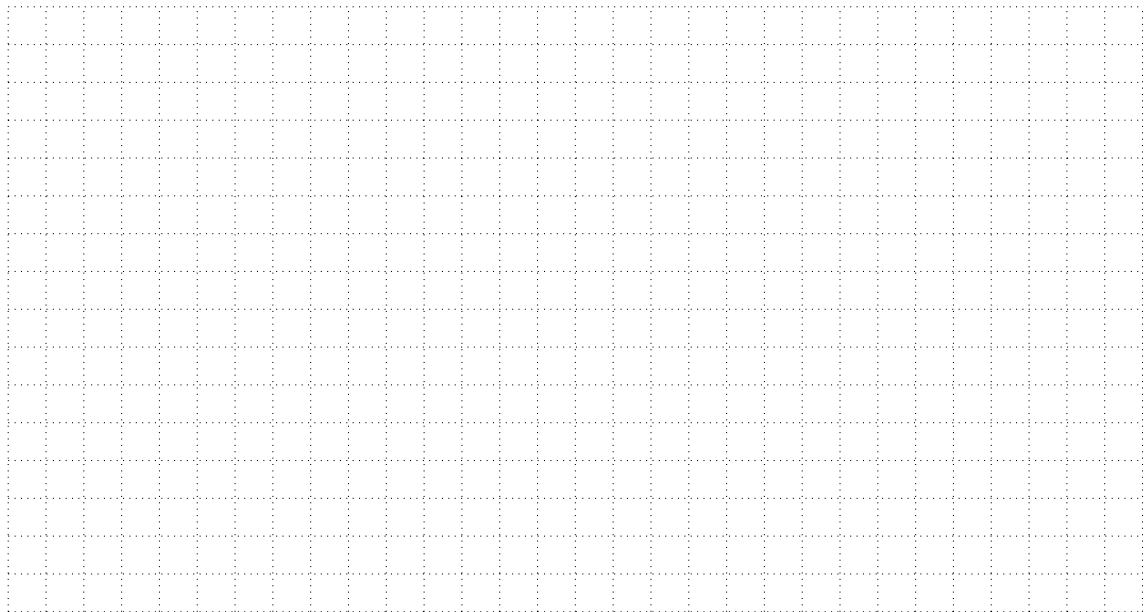


Dir stehen hierfür die Bausteine `gehe nach rechts`, `gehe nach links`, `gehe nach oben`, `fange Fisch`, `liefere Fisch ab`, `auf Fisch`, `auf Insel`, `wiederhole x mal` und `falls x mache y` zur Verfügung.



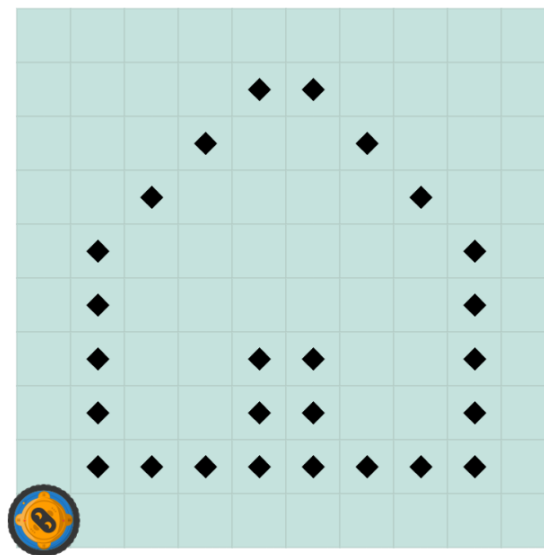
Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript:



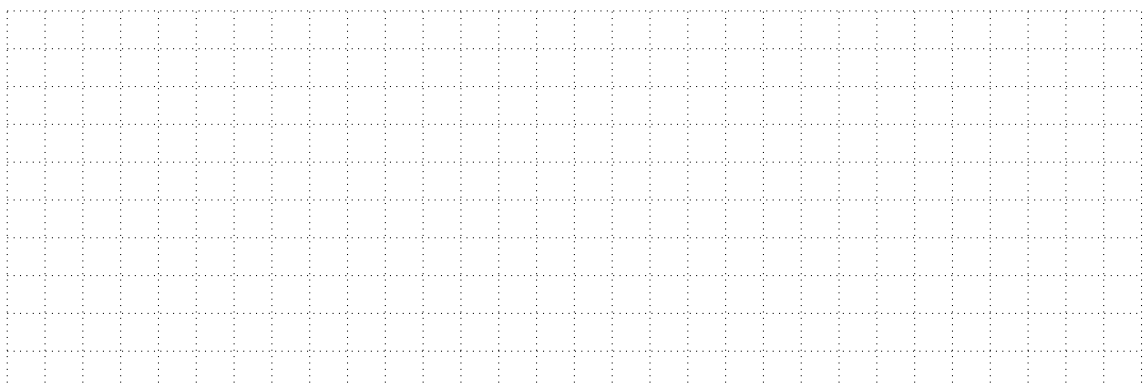


(d) Haus malen

Programmiere den Roboter: Der Roboter soll alle markierten Felder färben:



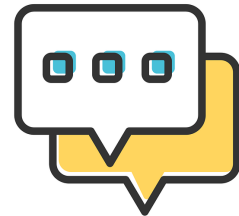
Dir stehen hierfür die Bausteine `gehe nach rechts`, `gehe nach links`, `gehe nach oben`, `gehe nach unten`, `färbe das Feld`, `auf Markierung`, `wiederhole x mal` und `falls x mache y` zur Verfügung.





### 2.3 Wiederholung mit Bedingung

Bisher wussten wir bei Wiederholungen bereits deren genaue Anzahl. Meistens ist das aber nicht der Fall. Stattdessen sollen Wiederholungen solange ausgeführt werden, bis irgendein Ereignis eintritt. Dann soll die Wiederholung gestoppt werden. Dieses Konzept nennt man **Wiederholung mit Bedingung**.



#### Merke (Bedingte Wiederholung)

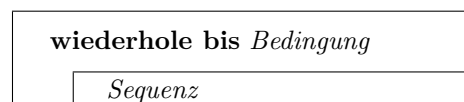
Die Kontrollstruktur **bedingte Wiederholung** wird verwendet, um eine Sequenz abhängig von einer Bedingung auszuführen. Sie wird dazu verwendet, Anweisungen oder ganze Sequenzen zu wiederholen, ohne dass zuvor die Anzahl der Wiederholungen bekannt sein muss.

Die Anweisungen werden dann solange ausgeführt, bis eine Abbruchbedingung eintritt (z.B. eine gedrückte Taste, ein Hindernis, usw.).



Bedingte Wiederholung

#### Struktogramm:



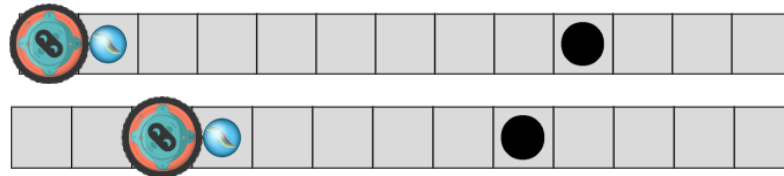


**Arbeitsauftrag 5: Bedingte Wiederholung**

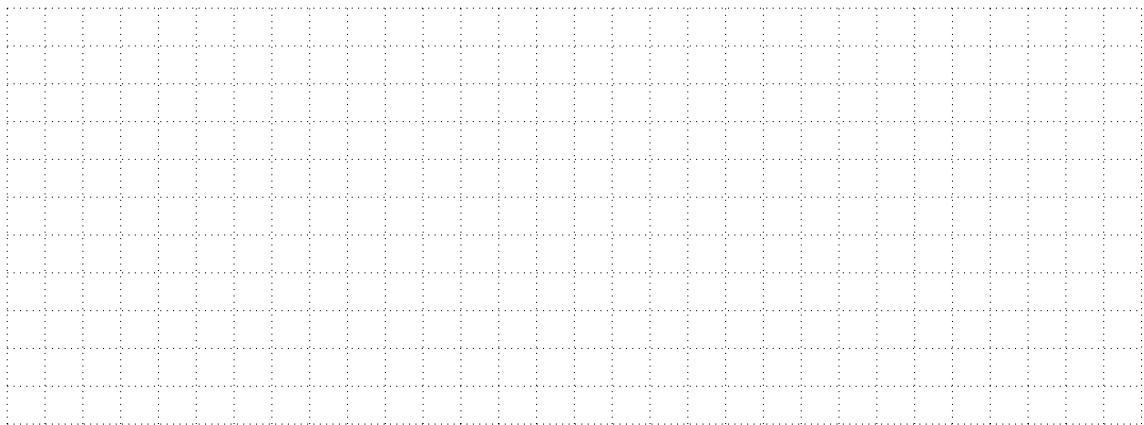
Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu **Arbeitsheft Algorithmik**.

(a) Murmel ablegen

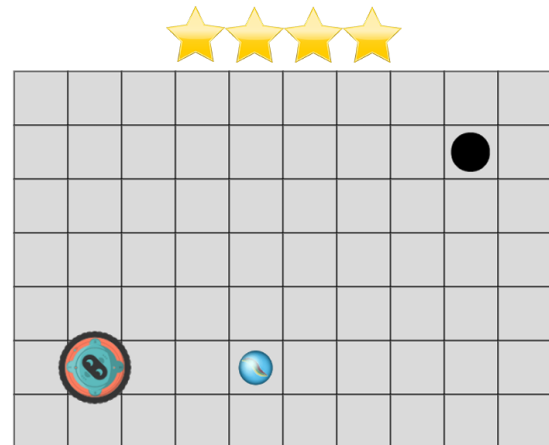
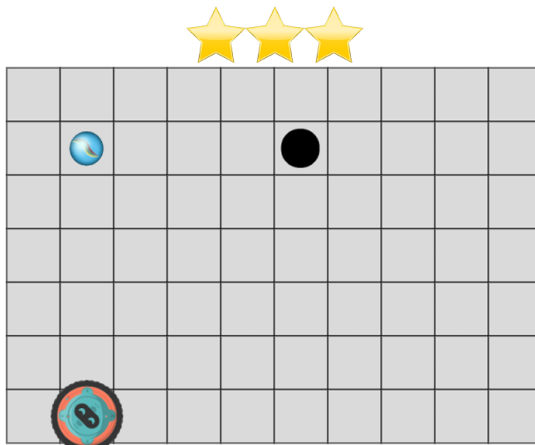
Programmiere den Roboter: Der Roboter soll die Murmel im Loch ablegen. Die Murmel liegt immer vor dem Roboter, das Loch immer einige Schritte dahinter. *Dein Programm muss mit allen Testfällen zurechtkommen.*

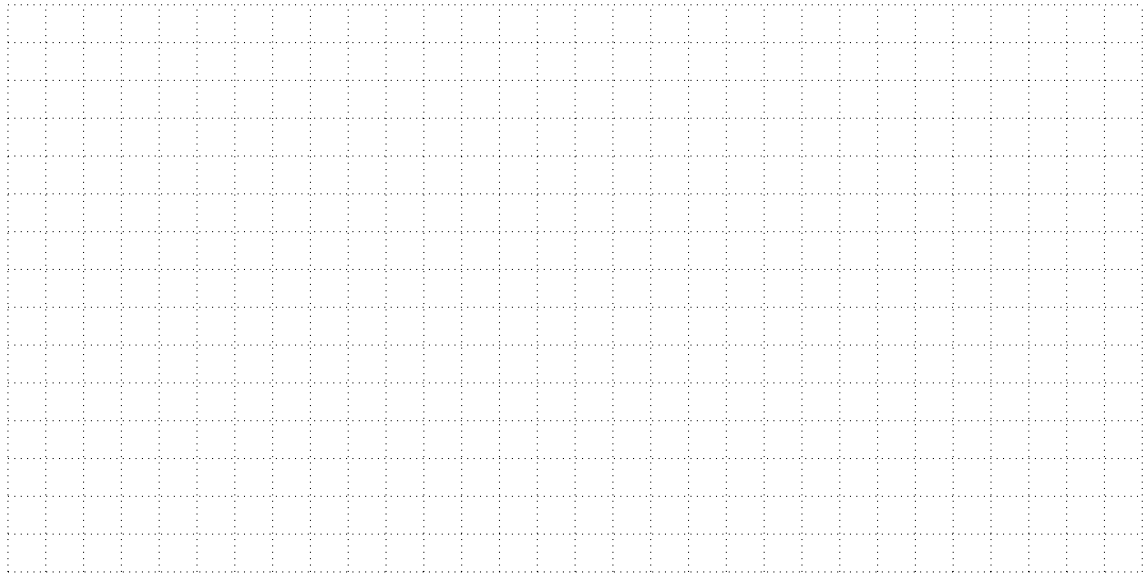


Dir stehen hierfür die Bausteine **gehe nach rechts**, **hebe Murmel auf**, **lege Murmel ab**, **auf Loch**, **auf Murmel**, **wiederhole bis x**, **wiederhole solange x** und **nicht** zur Verfügung.



Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Bei der 3-Sterne-Version gibt es zusätzlich den Baustein **gehe nach oben**, bei der 4-Sterne-Version nochmals zusätzlich die Bausteine **gehe nach links** und **links vom Gitterrand**. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript. *Beachte auch hier, dass dein Programm jeweils mit allen Testfällen zurechtkommen muss:*

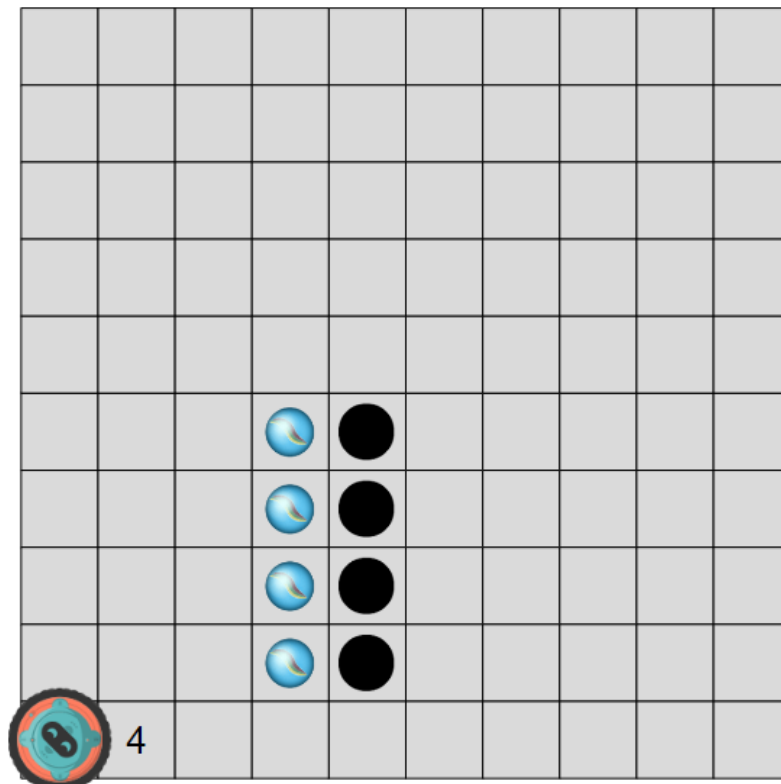




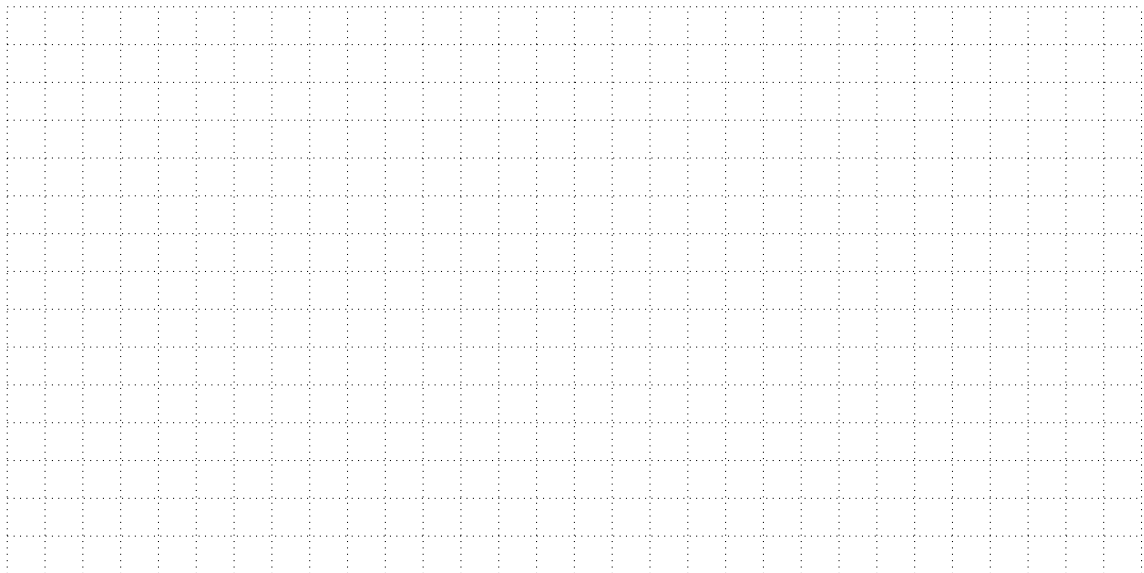
(b) Alle Murmeln (anspruchsvoll)

Diese Aufgabe findest du unter Variablen und Logik - Übungen.

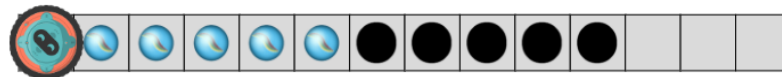
Programmiere den Roboter: Der Roboter soll alle Murmeln jeweils in einem Loch ablegen. Die Anzahl der Murmeln steht im Feld neben dem Roboter. Wenn er darauf steht, kann er die „Zahl auf dem Feld“ lesen:



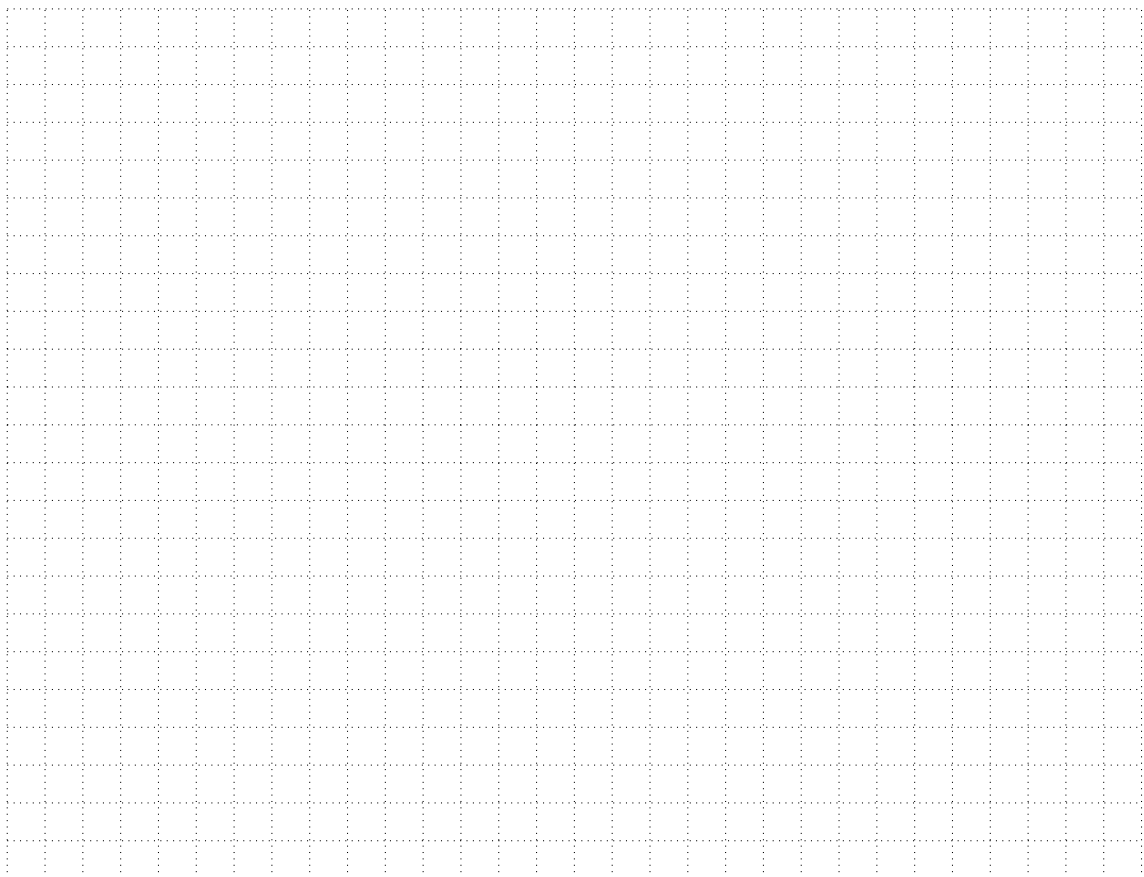
Dir stehen hierfür die Bausteine `gehe nach rechts` , `gehe nach links` , `gehe nach oben` , `gehe nach unten` , `hebe Murmel auf` , `lege Murmel ab` , `Zahl auf dem Feld` , `wiederhole x mal` , `Anzahl Murmeln` , `setze Anzahl Murmeln auf` und `erhöhe Anzahl Murmeln um 1` zur Verfügung.



Zu dieser Aufgabe gibt es auch noch folgende **4-Sterne**-Version. Diese hat einen hohen Schwierigkeitsgrad! Versuche diese Aufgabe zu lösen und dokumentiere deine Programme hier im Skript:

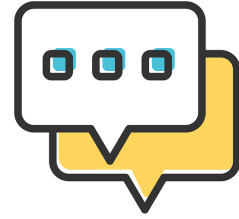


Dir stehen hierfür die Bausteine `gehe nach rechts` , `gehe nach links` , `gehe nach oben` , `gehe nach unten` , `hebe Murmel auf` , `lege Murmel ab` , `Zahl auf dem Feld` , `wiederhole x mal` , `wiederhole solange` , `Anzahl Murmeln` , `setze Anzahl Murmeln auf` , `Ziffer` und `erhöhe Anzahl Murmeln um x` zur Verfügung.



## 2.4 Zweiseitig bedingte Anweisung

Bisher haben wir uns bei Bedingungen nur darauf beschränkt, Anweisungen auszuführen, wenn eine zu überprüfende Bedingung **wahr** ist. Falls die Bedingung nicht eingetreten ist, wurden alle Anweisungen innerhalb der Kontrollstruktur ignoriert und das restliche Programm abgearbeitet. Oftmals ist es aber notwendig, dass auch im Falle einer nicht eintretenden Bedingung bestimmte Anweisungen ausgeführt werden sollen.



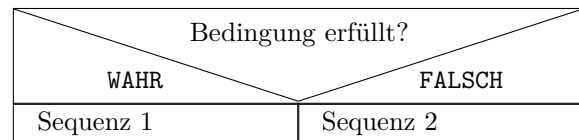
### Merke (Zweiseitig bedingte Anweisung)

Die Kontrollstruktur **zweiseitig bedingte Anweisung** dient dazu, um unterschiedliche Sequenzen abhängig von einer Bedingung auszuführen. Dabei werden sowohl im Fall **wahr** (engl. **true**) als auch im Fall **falsch** (engl. **false**) zusätzliche Anweisungen ausgeführt.



*Zweiseitig bedingte Anweisung*

### Struktogramm:

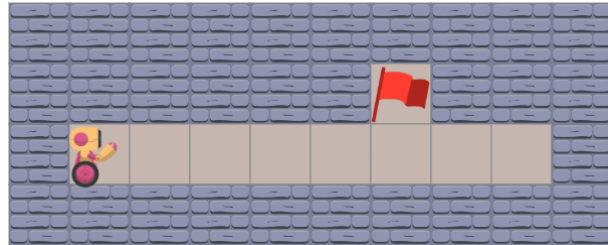
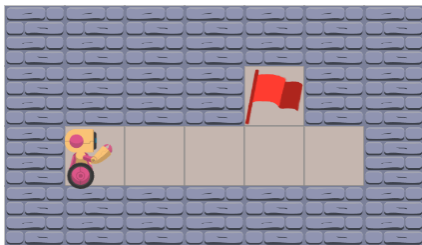


### Arbeitsauftrag 6: Zweiseitig bedingte Anweisung

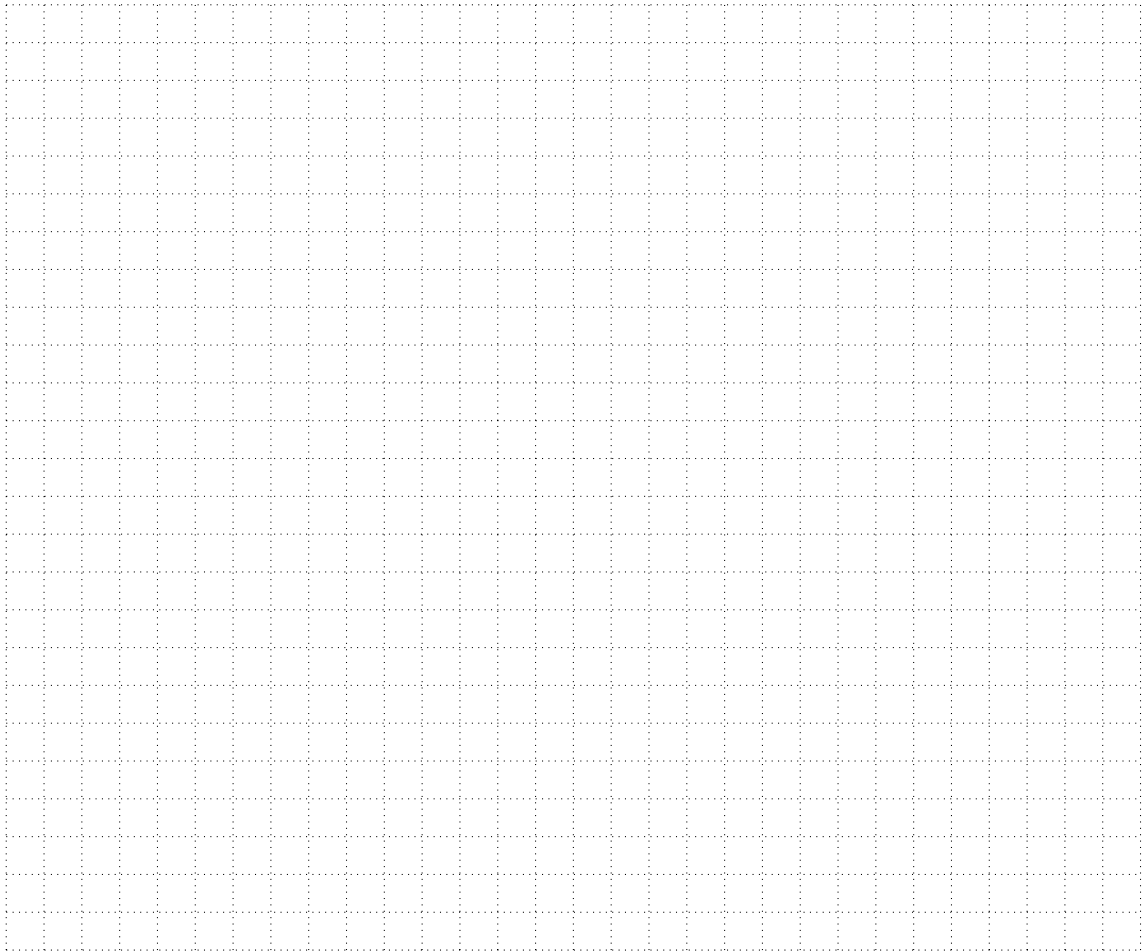
Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu **Arbeitsheft Algorithmen**.

#### (a) Finde das Ziel

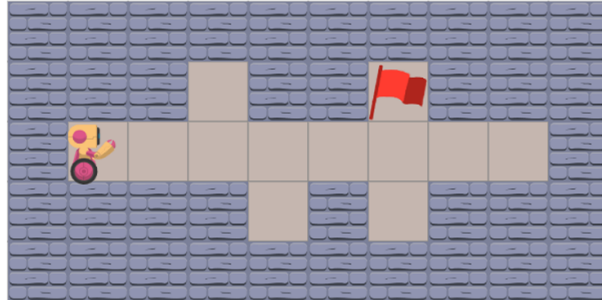
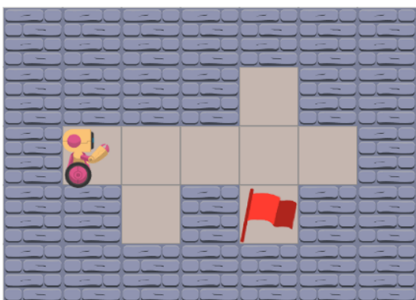
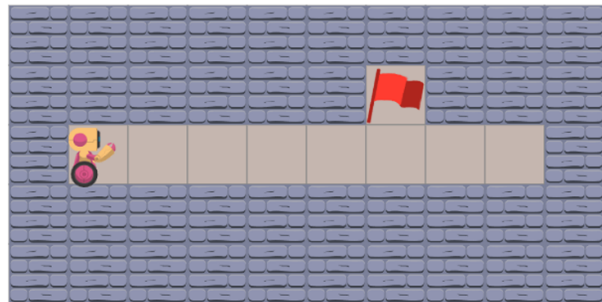
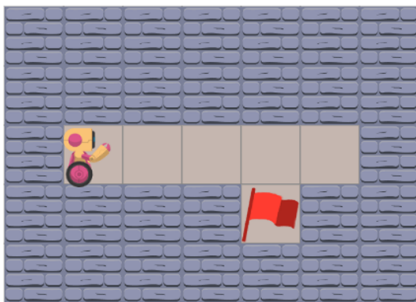
Programmiere Roboter: Der Roboter soll die rote Flagge erreichen. Die rote Flagge befindet sich immer in einer Lücke über dem Gang. *Beachte, dass dein Programm mit allen Testfällen zurechtkommen muss:*

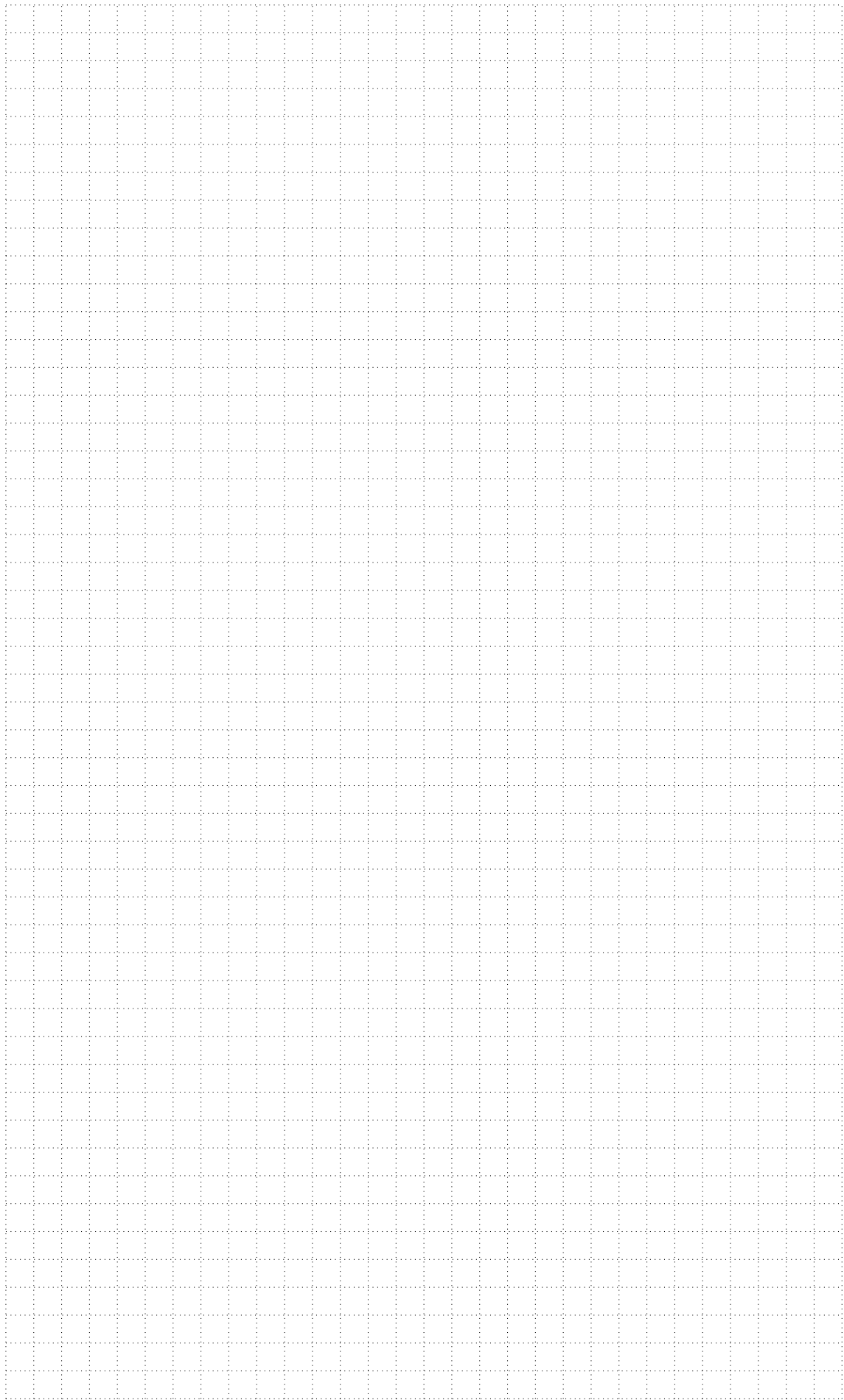


Dir stehen hierfür die Bausteine **gehe vorwärts**, **drehe nach rechts**, **drehe nach links**, **vor Hindernis**, **Flagge erobert**, **falls x mache y**, **falls x mache y sonst z**, **wiederhole bis x** und **wiederhole solange x** zur Verfügung.



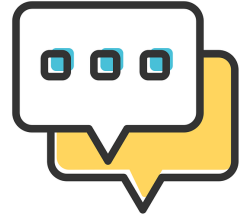
Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Hier steht dir zusätzlich noch der **nicht**-Baustein zur Verfügung, mit dem du beispielsweise rausfinden kannst, ob du auf keinem grünen Feld stehst. Löse auch diese Aufgaben und dokumentiere deine Programme hier im Skript. Beachte, dass dein Programm jeweils beide dargestellten Fälle richtig absolvieren muss:





## 2.5 Vermischte und schwere Aufgaben

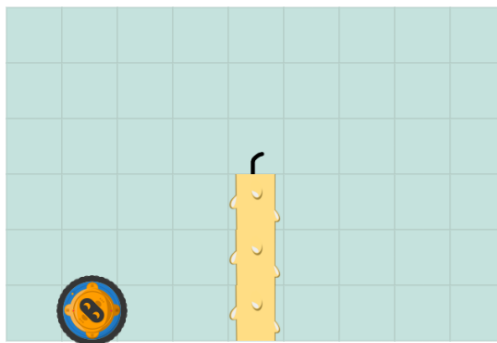
In diesem Abschnitt findest du nun Aufgaben, die du mit den bisher gelernten Kontrollstrukturen lösen kannst. Versuche, so viele Sterne wie möglich zu erreichen.



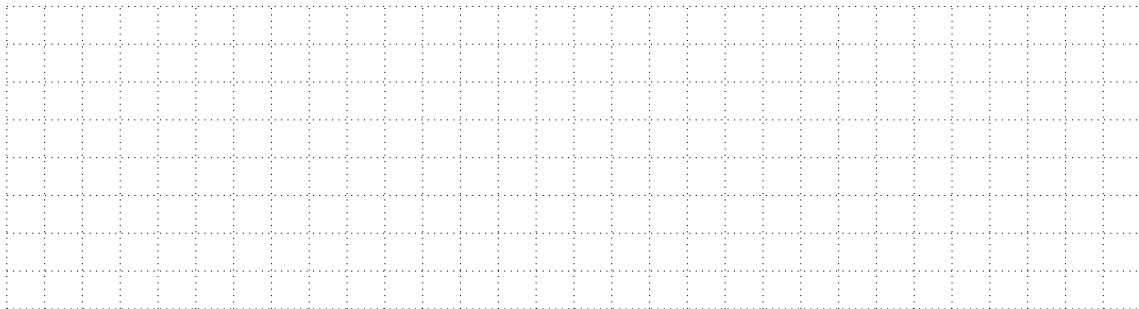
### Arbeitsauftrag 7: Zünde alle Kerzen an

Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu Arbeitsheft Algorithmik.

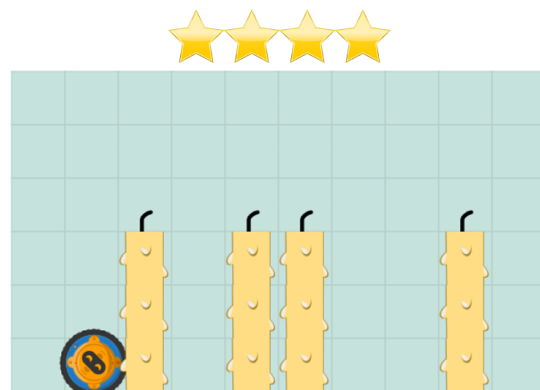
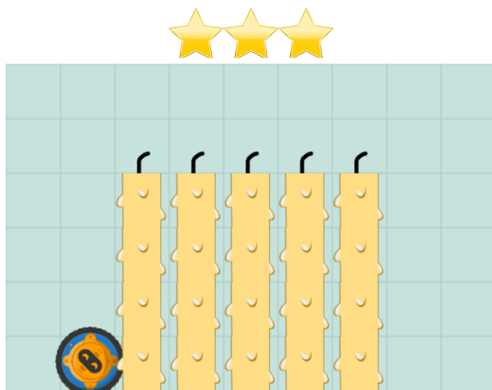
Der Roboter soll alle Kerzen anzünden:

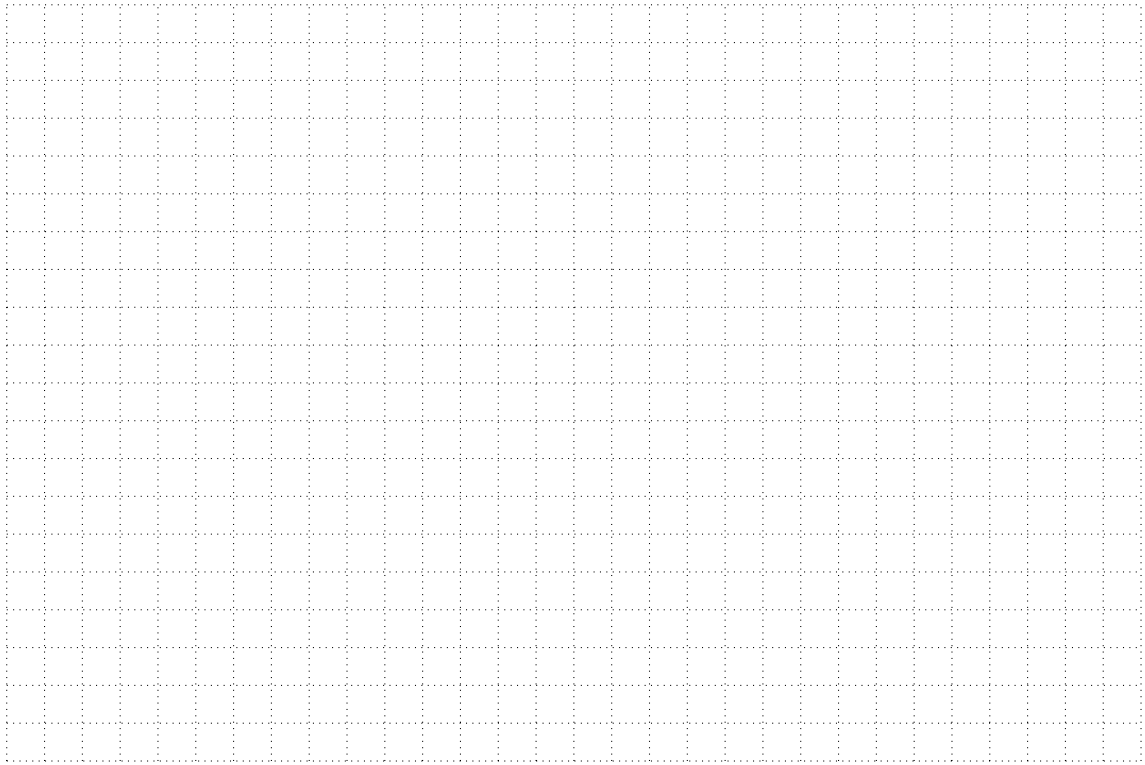


Dir stehen hierfür die Bausteine `gehe nach rechts`, `gehe nach links`, `gehe nach oben`, `gehe nach unten`, `zünde Kerze an` und `wiederhole x mal` zur Verfügung.



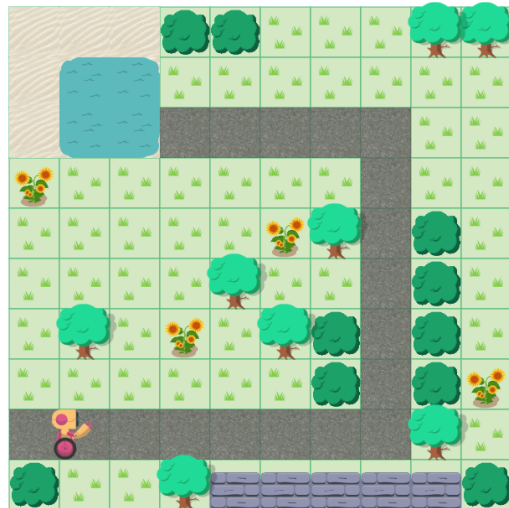
Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Bei der 4-Sterne-Variante stehen dir zusätzlich auch noch die Bausteine `auf Kerze` und `falls x mache y` zur Verfügung:



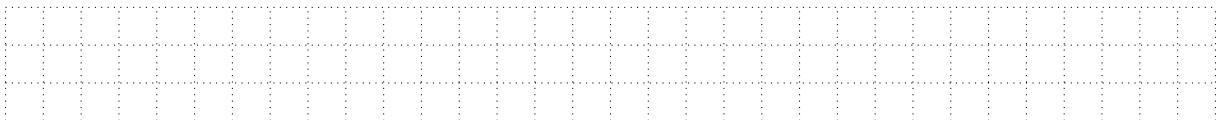


**Arbeitsauftrag 8: Finde den Weg zum See**

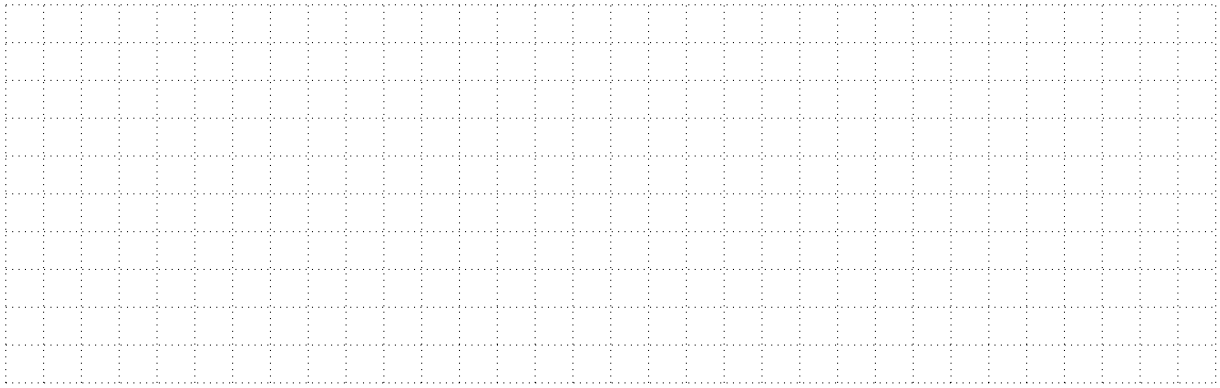
Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu **Arbeitsheft Algorithmen**.  
 Programmiere den Roboter: Der Roboter möchte schwimmen gehen. Auf dem Weg zum See darf er die Straße nicht verlassen:



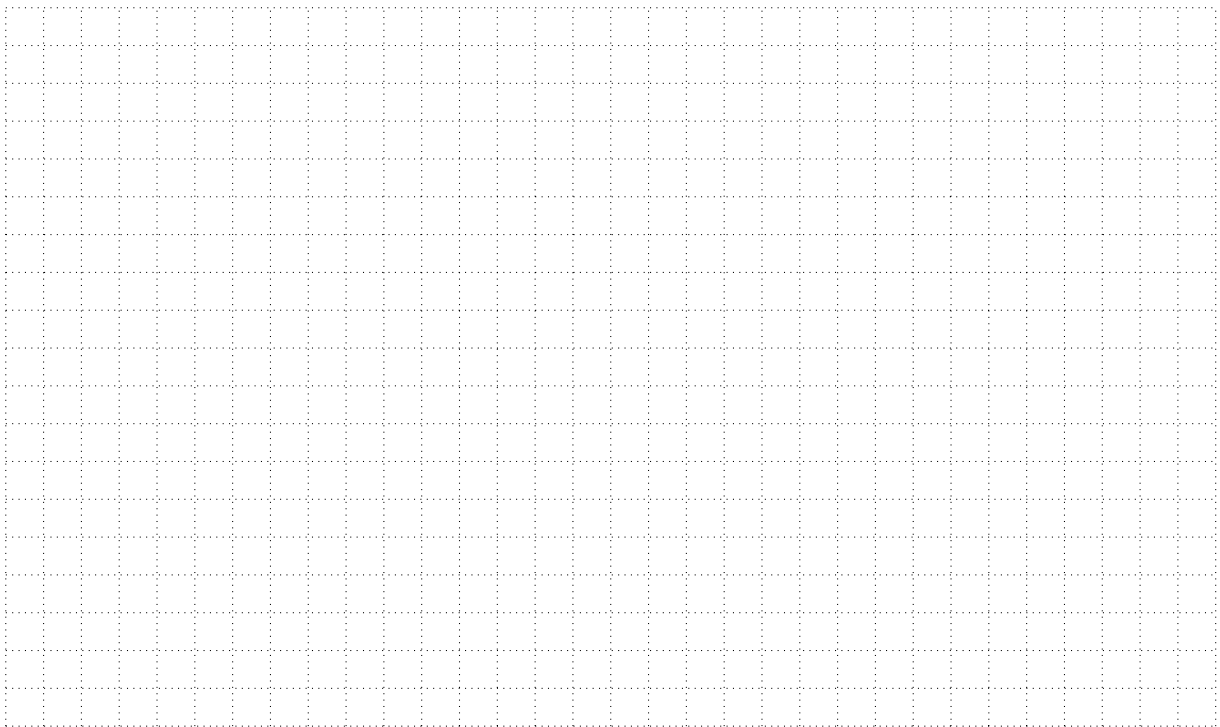
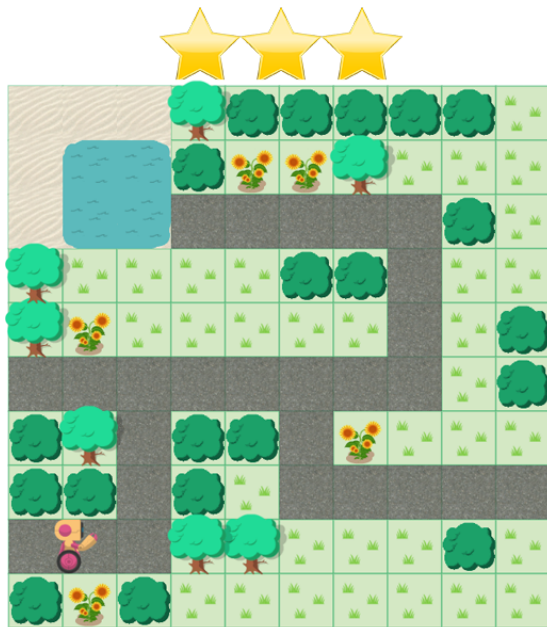
Dir stehen hierfür die Bausteine **drehe nach links** , **drehe nach rechts** , **gehe vorwärts** , **vor Hindernis** , **falls x mache y** und **wiederhole x mal** zur Verfügung.







Zu dieser Aufgabe gibt es auch noch eine **3-Sterne-** und **4-Sterne-**Version. Hier stehen dir zusätzlich noch die Bausteine `nicht` , `falls x mache y sonst z` zur Verfügung:

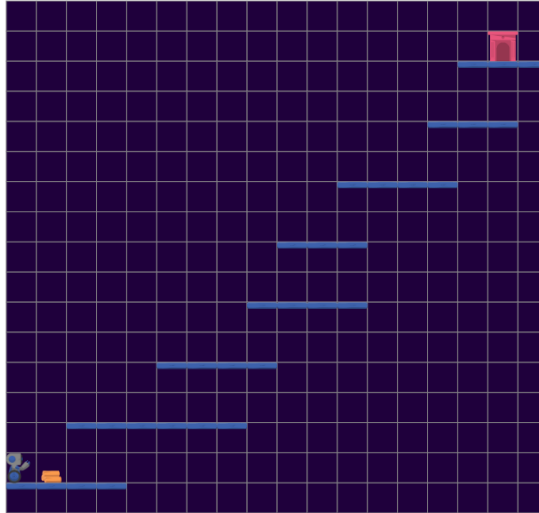


**Arbeitsauftrag 9: Holz und Plattformen**

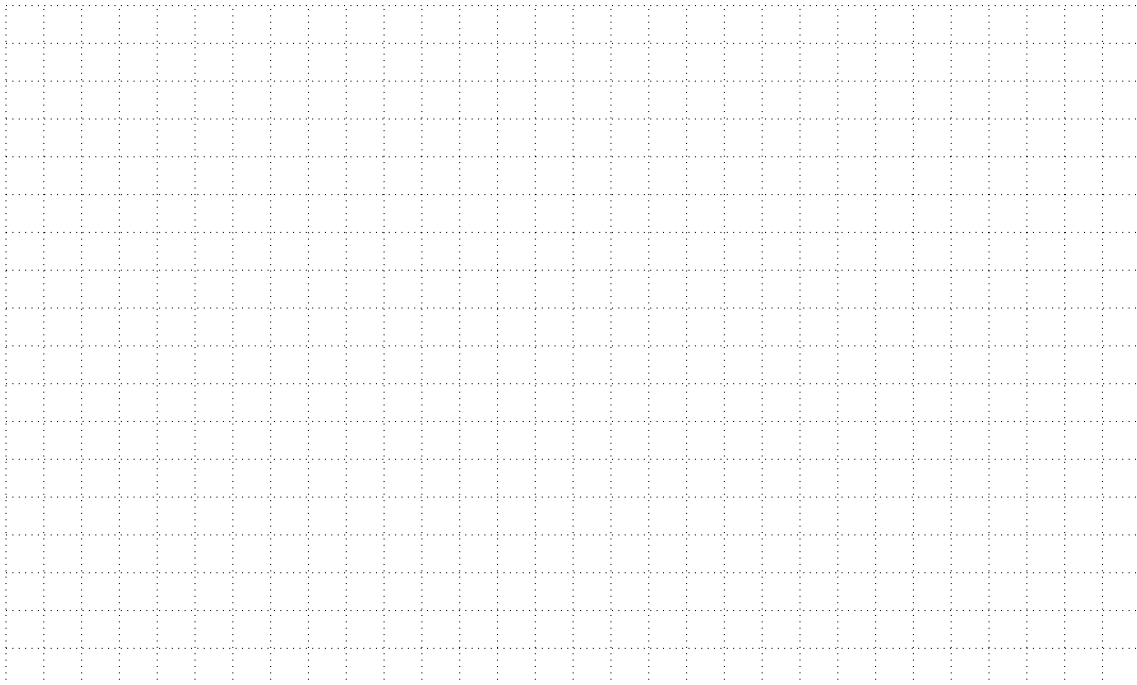
Logge dich auf der Seite [www.jwinf.de](http://www.jwinf.de) mit deiner Kennung ein und gehe zu **Arbeitsheft Algorithmen**.

**(a) 2-Sterne Variante:**

Programmiere den Roboter: Der Roboter soll das Holz im Kamin ablegen:

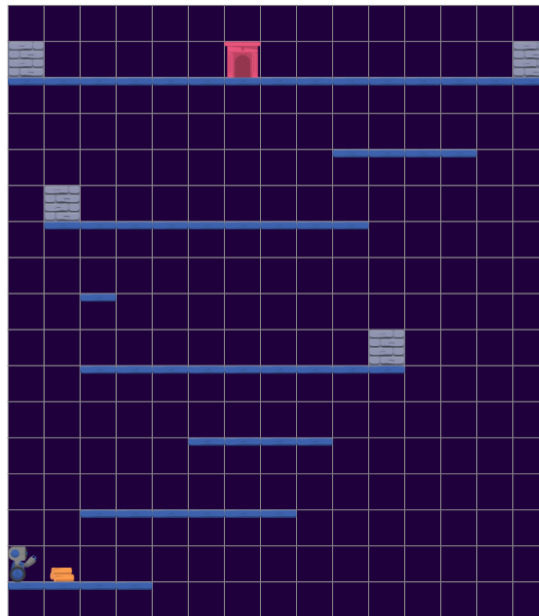


Dir stehen hierfür die Bausteine `gehe vorwärts`, `spring hoch`, `Holz einsammeln`, `Holz ablegen`, `Plattform oben`, `wiederhole x mal` und `falls x mache y` zur Verfügung.



**(b) 3-Sterne-Variante:**

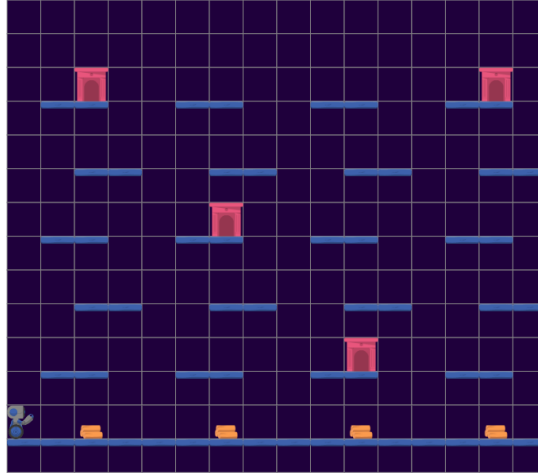
Programmiere den Roboter: Der Roboter soll das Holz im Kamin ablegen:



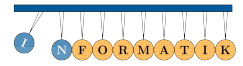
Dir stehen hierfür die Bausteine `gehe vorwärts` , `drehe um` , `spring hoch` , `Holz einsammeln` , `Holz ablegen` , `beim Kamin` , `Plattform oben` , `vor Hindernis` , `wiederhole x mal` , `falls x mache y` und `falls x mache y sonst z` zur Verfügung.

(c) 4-Sterne-Variante:

Programmiere den Roboter: Der Roboter soll jedes Holz in einem Kamin ablegen:



Dir stehen hierfür die Bausteine `gehe vorwärts` , `spring hoch` , `gehe rückwärts` , `drehe um` , `Holz einsammeln` , `Holz ablegen` , `beim Kamin` , `auf Holz` , `wiederhole x mal` , `falls x mache y` und `falls x mache y sonst z` zur Verfügung.



# Platz für Notizen

A large rectangular area filled with a grid of small, evenly spaced dotted lines, intended for students to take notes on.

