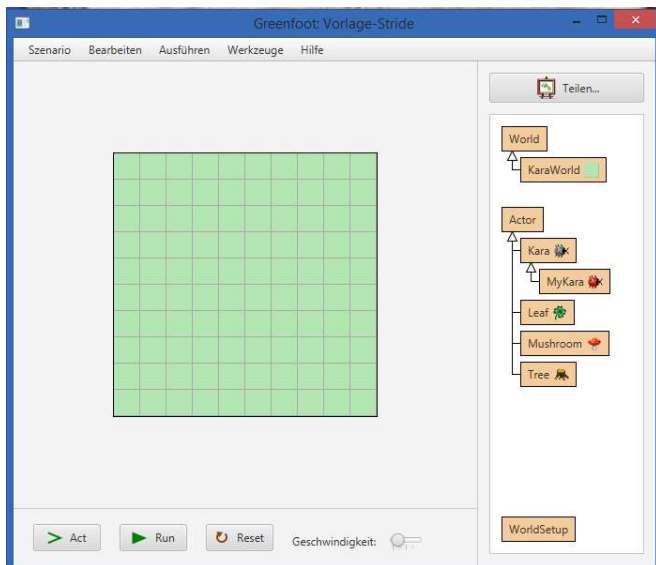


L1_1.1 Die Sequenz

Programme bestehen aus Code, der in Abschnitte unterteilt werden kann. Diese werden als Sequenz bezeichnet.

(I) Problemstellung

Öffnen Sie das Szenario "Vorlage-Stride" aus dem Ordner *Vorlagen_Szenarien*.



Vorgegeben ist eine Welt (grüne Fläche), auf der verschiedene Akteure (Käfer, Blatt, Pilz und Baum) platziert werden können.

Mit Hilfe des Kontextmenüs lässt sich ein neuer Käfer (MyKara) erzeugen und in die Welt setzen.

- 1 Erweitern Sie das Szenario "Vorlage-Stride" so, dass sich MyKara einen Schritt vorwärts bewegt.

Beachten Sie das Informationsmaterial

L1_1.1 Information Greenfoot Stride Programmumgebung.docx.

Alternativ zu dem Informationsmaterial können Sie zwei Erklärvideos verwenden.

Diese finden Sie unter folgenden Links:

- Einführung in Greenfoot-Kara (Teil 1), <https://vimeo.com/327596565>
- Einführung in Greenfoot-Kara (Teil 2), <https://vimeo.com/327571735>

Kodieren Sie die Lösung und speichern Sie Ihr Ergebnis unter dem Namen *L1_1.1_A1 Sequenz*.

- 2 Das Szenario "Vorlage-Stride" soll so erweitert werden, dass MyKara zwei Schritte geht, anschließend nach links dreht und dann drei Schritte geht.

Beachten Sie das Informationsmaterial *L1_1.2 Information Greenfoot Stride Sequenz.docx*.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_1.1_A2 Sequenz*.

L1_1.2 Übungsaufgaben zur Sequenz

Aufgabe 1



Welche Aktionen brauchen Sie, damit MyKara den Baum anschaut?

Erstellen Sie ein Struktogramm, damit MyKara den Baum anschaut und kodieren Sie die Lösung. Verwenden Sie das Szenario "Vorlage-Stride" und speichern Sie Ihr Ergebnis unter dem Namen *L1_1.2_A1 Sequenz*.

Aufgabe 2

Setzen Sie zusätzlich einen Baum (tree) in die Welt. Mit welcher Aktion kann MyKara überprüfen, ob er vor einem Baum steht?

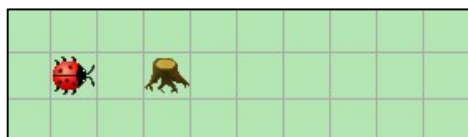
Aufgabe 3

Was passiert, wenn Sie MyKara mittels der Aktion *move()* in einen Baum laufen lassen?

Aufgabe 4

Gestalten Sie das Szenario "Vorlage-Stride" so, dass MyKara vor einem Baum steht. Diesen soll er dann umgehen.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_1.2_A4 Sequenz*.



Aufgabe 5

Erstellen Sie folgende Welt:

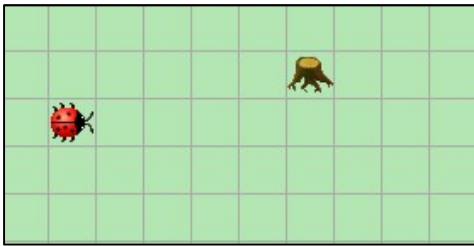


Entwickeln Sie eine Lösung, die MyKara zum Kleeblatt führt, indem er immer wieder den Weg zwischen den Bäumen wählt (siehe Abb.). Beim Kleeblatt angekommen, soll er dieses aufheben.

Erstellen Sie ein Struktogramm und kodieren Sie die Lösung.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_1.2_A5 Sequenz*.

Aufgabe 6



MyKara soll zu dem Baum gehen und vor ihm stehen bleiben. Dazu wurde der folgende Programmcode entwickelt.

```
public void act() // overrides method in Actor
{
    turnRight()
    move()
    turnLeft()
    move()
    move()
    move()
}
```

Beschreiben Sie die Wirkungsweise des Programmcodes und beurteilen Sie, ob MyKara sein Ziel erreicht.

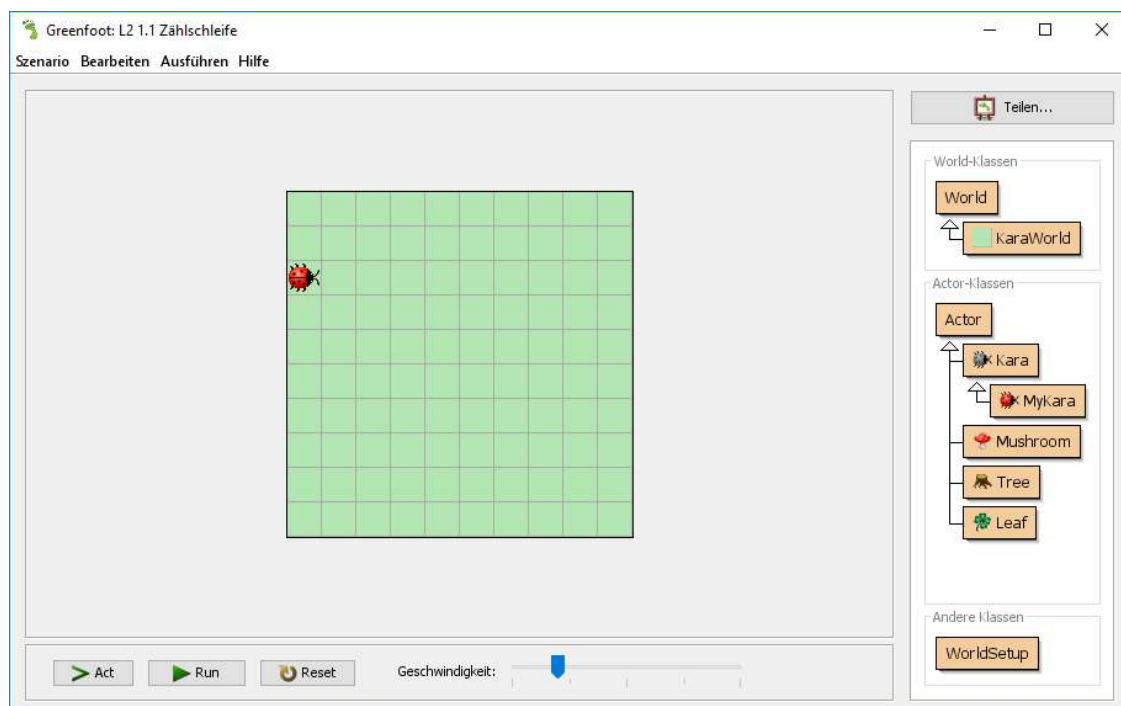
L1_2.1 Die Zählerschleife (for-Schleife)

(I) Problemstellung

Öffnen Sie das Szenario "Vorlage-Stride" aus dem Ordner *Vorlagen_Szenarien*.

Erzeugen Sie einen neuen Käfer MyKara und platzieren Sie ihn auf der vorgegebenen Welt (siehe Abb.).

MyKara soll nun neun Schritte gehen, so dass er am Ende auf der anderen Seite steht:



J1	BPE 5: Grundlagen der Programmierung Arbeitsauftrag	Informatik
----	---	------------

1 Überlegen Sie:

- Wie lösen Sie das Problem mit den bisher bekannten Mitteln?
- Welchen Nachteil hat diese Vorgehensweise, wenn die Kara-Welt 100 x 100 Felder hätte?
- Entwickeln Sie einen geeigneten Vorschlag zur Lösung des Problems.

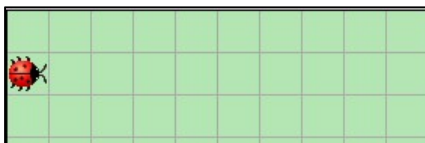
2 Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_2.1 Zählerschleife*.

Beachten Sie das Informationsmaterial *L1_2 Information for-Schleife.docx*.

L1_2.2 Übungsaufgaben zur for-Schleife – Teil 1

Öffnen Sie zur Bearbeitung der folgenden Aufgabenstellungen jeweils das Szenario "Vorlage-Stride".

Aufgabe 1



MyKara macht neun Schritte und legt dabei jedes Mal ein Blatt ab, so dass eine Art „Blattweg“ vom linken zum rechten Spielfeldrand entsteht.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_2.2_A1 Lösung*.

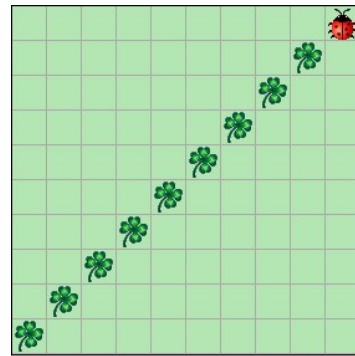
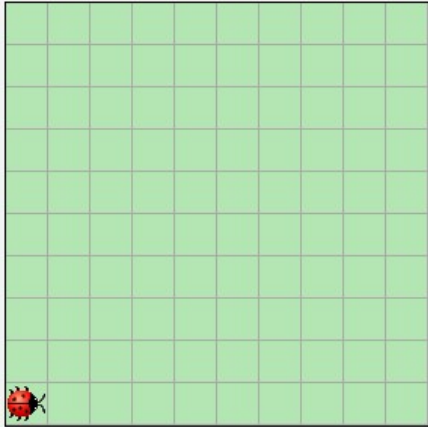


Aufgabe 2

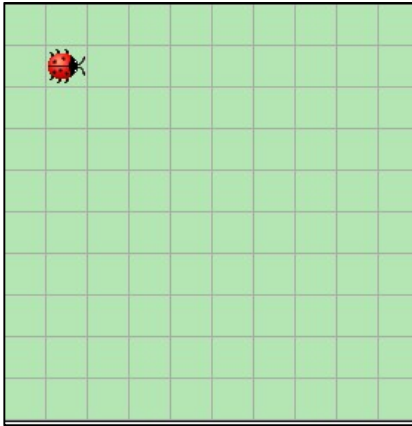
MyKara soll auf dem Spielfeld die Diagonale von links unten nach rechts oben mit Blättern auslegen. In der Startposition steht MyKara in der unteren linken Ecke.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_2.2_A2 Lösung*.

J1	BPE 5: Grundlagen der Programmierung Arbeitsauftrag	Informatik
----	---	------------

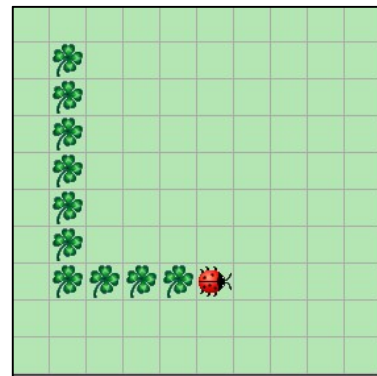


Aufgabe 3



MyKara soll mit Blättern ein "L" schreiben, wobei der lange Balken des "L" 7 Felder und der kurze Balken 4 Felder lang sein soll. Die Aufgabe soll mit 2 aufeinander folgenden Schleifen gelöst werden.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_2.2_A3 Lösung*.



Aufgabe 4

```
public void act() overrides method in Actor
{
    for each ( var zaehler in 1 .. 6 )
    {
        turnLeft()
        move()
        turnLeft()
        move()
        turnLeft()
        move()
        turnLeft()
        move()
    }
}
```

In einer Kara-Welt wurde der abgebildete Programmcode entwickelt.

- 4.1 Erläutern Sie den syntaktischen Fehler, den dieser Programmcode enthält.
- 4.2 Beschreiben Sie die Wirkungsweise des Programmcodes, nachdem der syntaktische Fehler korrigiert wurde.

L1 1.1 L1_2.3 Übungsaufgaben zur for-Schleife – Teil 2

Ergänzen Sie in den Lösungstabellen der folgenden Aufgabenstellungen die fehlenden Werte:

Aufgabe 1:

```
for each(int zaehler in 1..5)
{
    System.out.println ( zaehler );
}
```

Der Befehl `System.out.println (Variablenname)` bewirkt die Ausgabe des Variableninhalts am Bildschirm (hier: Inhalt der Variable `zaehler`).

Wert "zaehler"	Nr. des Durchlaufs	Ausgabe auf Bildschirm

Aufgabe 2:

```
for each( int i in 0..4)
{
    System.out.println ( i );
}
```

Wert "i"	Nr. des Durchlaufs	Ausgabe auf Bildschirm

Aufgabe 3:

act()
Wiederhole von i= 0 solange i < 10, Schrittweite 2
Ausgabe: i

Wert "i"	Bedingung erfüllt?	Nr. des Durchlaufs	Ausgabe auf Bildschirm

Aufgabe 4:

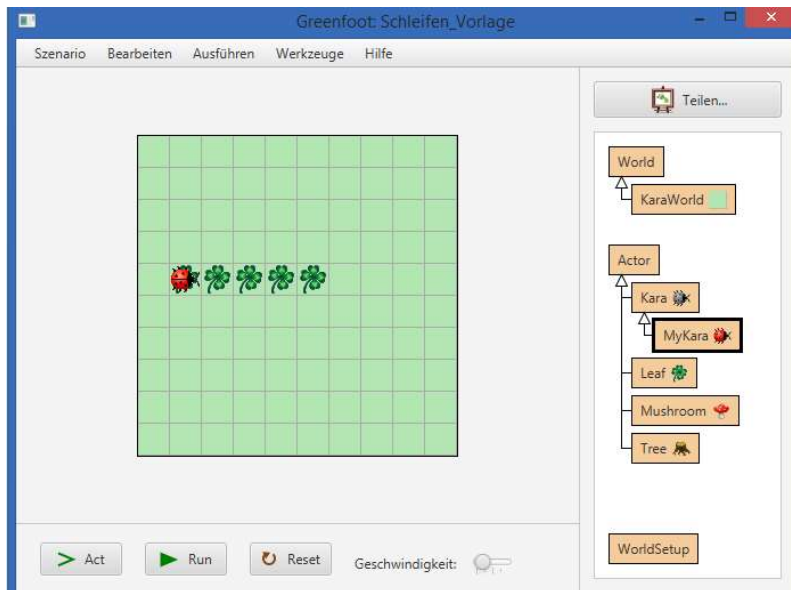
act()
Wiederhole von i= 0 solange i <= 10, Schrittweite 4
Ausgabe: i

Wert "i"	Bedingung erfüllt?	Nr. des Durchlaufs	Ausgabe auf Bildschirm

L1_3.1 Die while-Schleife

(I) Problemstellung

Öffnen Sie das Szenario "Vorlage-Stride" aus dem Ordner *Vorlagen_Szenarien* und platzieren Sie einen neuen Käfer und eine Reihe Kleeblätter in der Welt (siehe Abb.).



MyKara steht auf einem Kleeblatt und soll so lange geradeaus gehen, bis er nicht mehr auf einem Blatt steht.

1 Überlegen Sie:

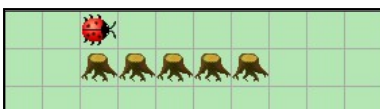
- Wie können Sie das Problem mithilfe der bisher kennengelernten Programmstrukturen lösen?
- Funktioniert Ihre Lösung – ohne Veränderungen am Programm vorzunehmen – auch dann, wenn MyKara 3 (oder 5, oder 8 oder 10) Kleeblätter vor sich hat?
- Wie könnte ein Programm aussehen bzw. was müsste eine Programmstruktur können, damit Sie das Problem lösen können?

2 Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_3.1 While-Schleife*.

Beachten Sie das Informationsmaterial *L1_3 Information while-Schleife.docx*.

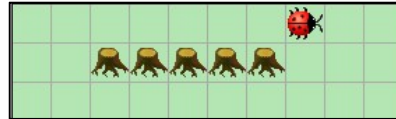
L1_3.2 Übungsaufgaben zur while-Schleife

Aufgabe 1



MyKara soll den Bäumen entlang gehen, bis rechts von ihm kein Baum mehr ist.

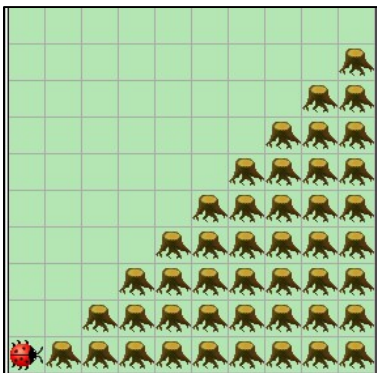
J1	BPE 5: Grundlagen der Programmierung Arbeitsauftrag	Informatik
----	---	------------



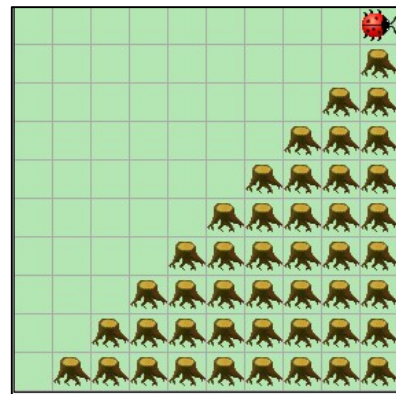
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_3.2_A1 While-Schleife* als Vorlage.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_3.2_A1 Lösung*.

Aufgabe 2



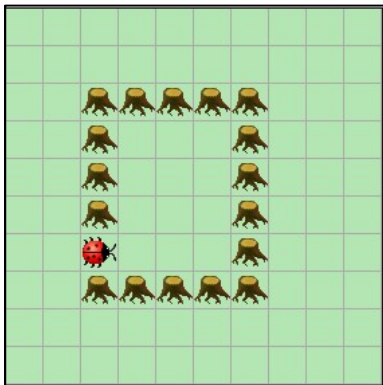
MyKara soll eine beliebig lange Treppe hochlaufen.



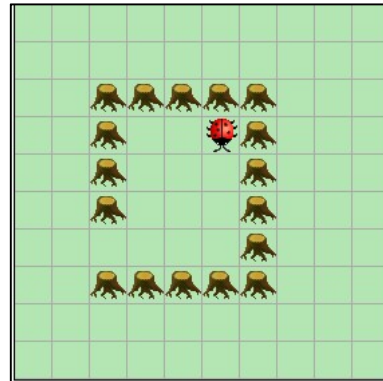
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_3.2_A2 While-Schleife* als Vorlage.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_3.2_A2 Lösung*.

Aufgabe 3



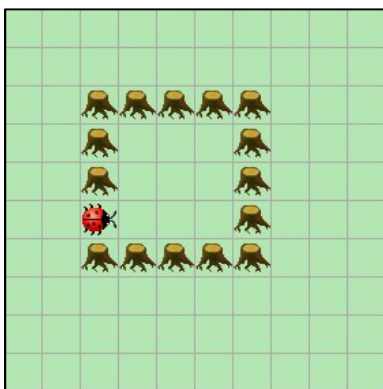
MyKara steht im Eingang seines rechteckigen Baus. MyKara soll in die obere hintere Ecke des Baus laufen (siehe Abb.). Beachten Sie, dass der Karabau verschiedene Ausmaße haben kann.



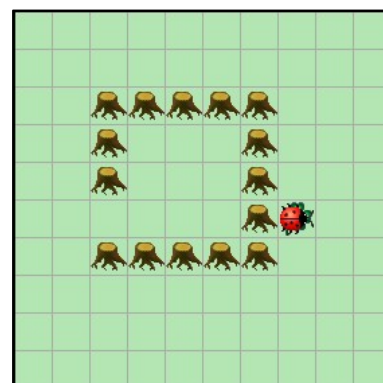
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_3.2_A3 While-Schleife* als Vorlage.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_3.2_A3 Lösung*.

Aufgabe 4



MyKara steht im Eingang seines rechteckigen Baus. MyKara soll im Uhrzeigersinn um den Bau laufen und an der gegenüberliegenden Seite ein Blatt ablegen. MyKara soll dort dieselbe Blickrichtung einnehmen wie in seiner Startposition.



Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_3.2_A4 While-Schleife* als Vorlage.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_3.2_A4 Lösung*.

Aufgabe 5

```
public void act() // overrides method in Actor
{
    var int zaehler = 0
    while ( zaehler < 10 )
    {
        move(3)
        turnLeft( )
        move(3)
        turnLeft( )
        move(3)
        turnLeft( )
        move(3)
        turnLeft( )
    }
}
```

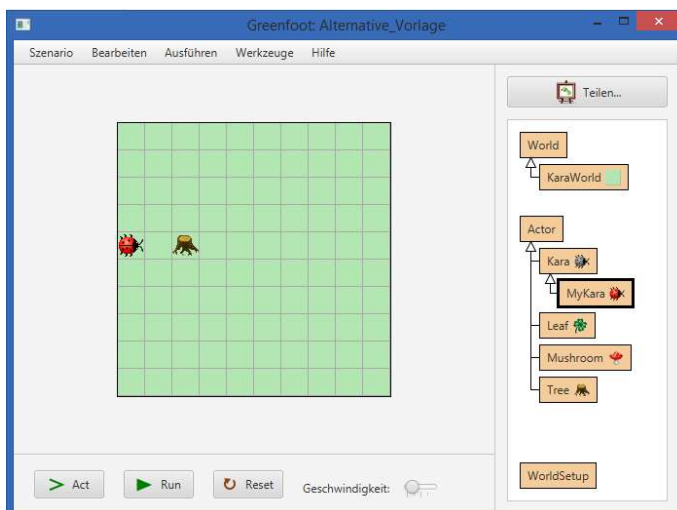
In einer Kara-Welt wurde der abgebildete Programmcode entwickelt.

- 5.1 Erläutern Sie den logischen Fehler, den dieser Programmcode enthält.
- 5.2 Beschreiben Sie die Wirkungsweise des Programmcodes, nachdem der Fehler korrigiert wurde.

L1 1.2 L1_4.1 Die Alternative (if-Anweisung)

(I) Problemstellung

Öffnen Sie das Szenario "Vorlage-Stride" aus dem Ordner *Vorlagen_Szenarien* und platzieren Sie – wie abgebildet – einen Käfer und einen Baum in der Welt.



MyKara soll einen Schritt machen und dann prüfen, ob ein Baum im Weg steht. Ist dies der Fall, soll MyKara den Baum rechts umgehen. Hinter dem Baum soll MyKara wieder zum Stehen kommen und in die gleiche Richtung schauen wie in der Startposition. Ist dies nicht der Fall, soll MyKara nichts tun.

1 Überlegen Sie:

- Welche unterschiedlichen Aktionen sind zu beachten?
- In welcher Reihenfolge soll MyKara die Aktionen durchführen?

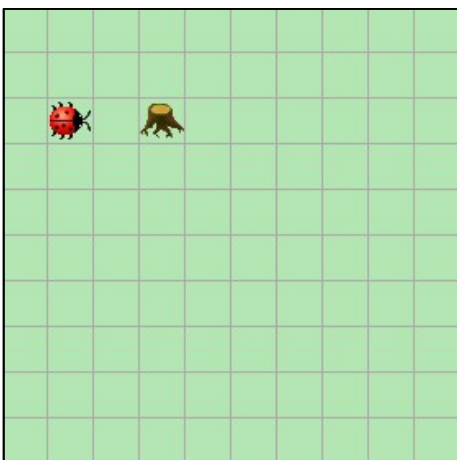
J1	BPE 5: Grundlagen der Programmierung Arbeitsauftrag	Informatik
----	---	------------

- 2 Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_4.1 Alternative*.

Beachten Sie das Informationsmaterial *L1_4 Information Alternative.docx*.

L1 1.3 L1_4.2 Übungsaufgaben zur if-Anweisung

Aufgabe 1



MyKara soll einen Schritt vorwärts gehen und prüfen, ob er vor einem Baum steht. Ist dies der Fall, soll er ein Blatt ablegen, den Baum links umgehen und in die gleiche Richtung schauen wie in der Startposition. Ist dies nicht der Fall, soll er nichts tun.



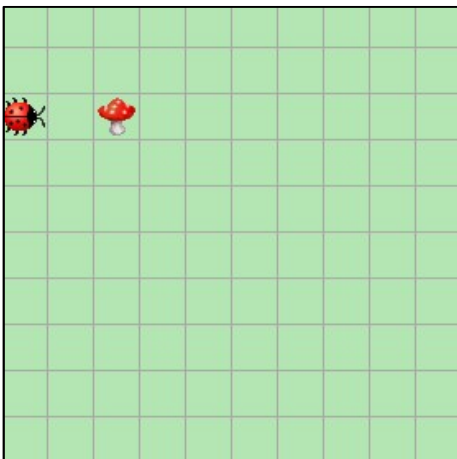
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_4.2_A1 Alternative* als Vorlage.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

Testen Sie Ihr Ergebnis, indem Sie den Standort von MyKara variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_4.2_A1 Lösung*.

Aufgabe 2



MyKara hat Angst vor Pilzen (Mushroom). Nachdem er einen Schritt vorwärts gemacht hat, soll er prüfen, ob vor ihm ein Pilz steht. Ist dies der Fall, soll er sich nach rechts drehen und drei Schritte vorwärts gehen. Ist dies nicht der Fall, soll er nur einen Schritt vorwärts gehen.



J1	BPE 5: Grundlagen der Programmierung Arbeitsauftrag	Informatik
----	---	------------

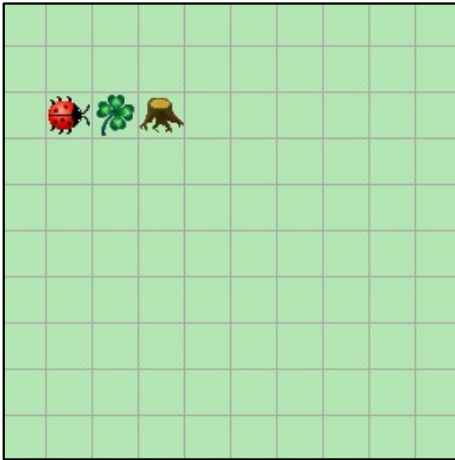
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_4.2_A2 Alternative* als Vorlage.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

Testen Sie Ihr Ergebnis, indem Sie den Standort von MyKara variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_4.2_A2 Lösung*.

Aufgabe 3



Nachdem MyKara einen Schritt vorwärts gegangen ist, soll er prüfen, ob sich vor ihm ein Baum befindet und ob er auf einem Blatt steht. Ist beides der Fall, soll er das Blatt aufheben, nach links drehen und einen Schritt vorwärts gehen. Ist beides nicht der Fall, soll er nach rechts drehen und einen Schritt vorwärts gehen.



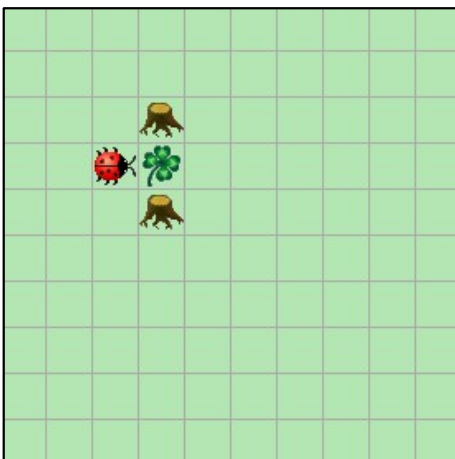
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_4.2_A3 Alternative* als Vorlage.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

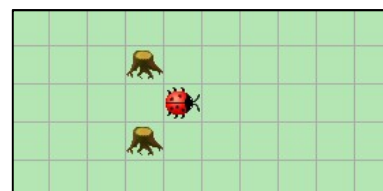
Testen Sie Ihr Ergebnis, indem Sie die Standorte der Akteure variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_4.2_A3 Lösung*.

Aufgabe 4



Nachdem MyKara einen Schritt vorwärts gegangen ist, soll er prüfen, ob links und rechts von ihm ein Baum steht. Ist dies der Fall, soll er prüfen, ob er auf einem Blatt steht. Wenn dies der Fall ist, soll er das Blatt aufheben und einen Schritt vorwärts gehen. Wenn er nicht auf einem Blatt steht, soll er ein Blatt ablegen und einen Schritt vorwärts gehen. Befinden sich neben MyKara keine zwei Bäume, soll er drei Schritte vorwärts gehen.



Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_4.2_A4 Alternative* als Vorlage.

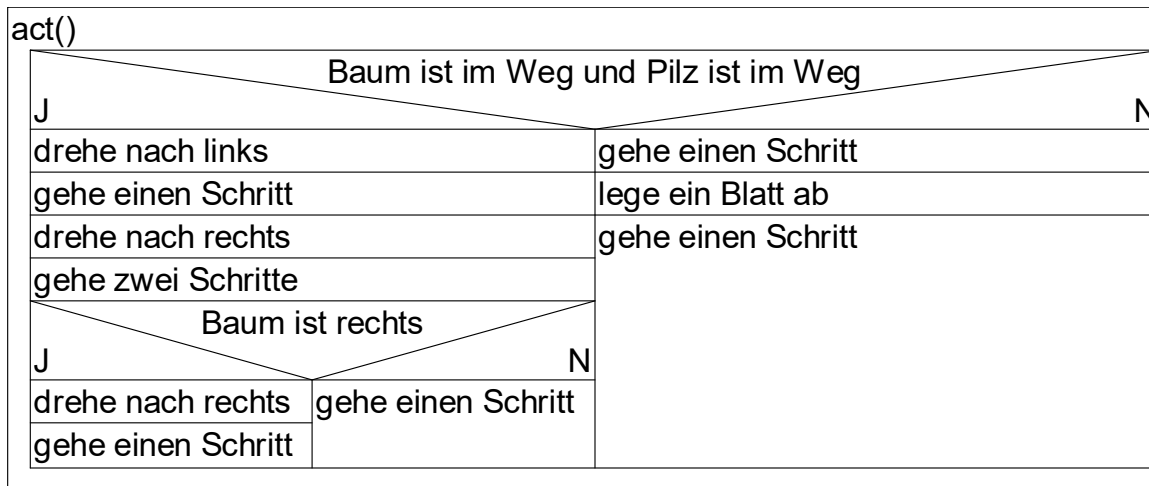
Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

Testen Sie Ihr Ergebnis, indem Sie die Standorte der Akteure variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_4.2_A4 Lösung*.

Aufgabe 5

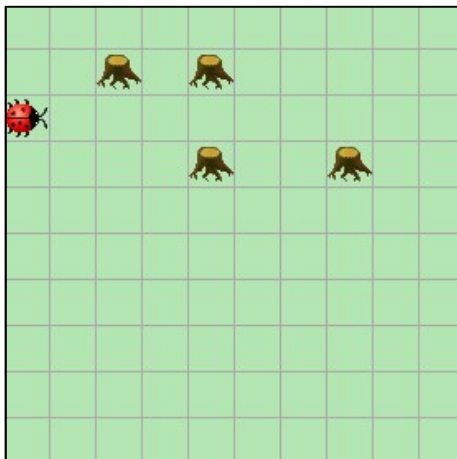
Für die Methode *act()* einer Kara-Welt wurde folgendes Struktogramm entwickelt:



Analysieren Sie das Struktogramm und erläutern Sie die zwei logischen Fehler.

L1_5.1 Übungsaufgaben zu Kontrollstrukturen

Aufgabe 1



MyKara soll neun Schritte über die Welt laufen und jedes Mal prüfen, ob links oder rechts von ihm ein Baum steht. Ist dies der Fall, soll er ein Blatt ablegen. Ist dies nicht der Fall, soll er ohne etwas zu tun einen Schritt vorwärts gehen.



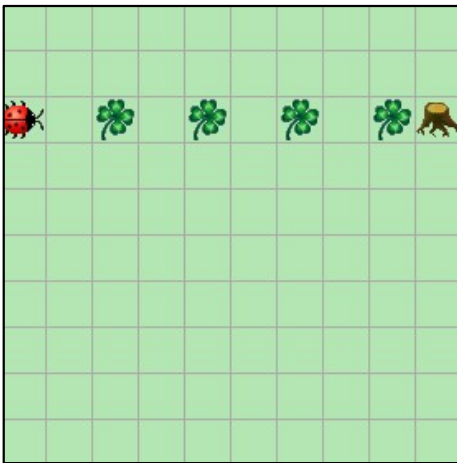
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_5.1_A1 Übung Kontrollstrukturen* als Vorlage.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

Testen Sie Ihr Ergebnis, indem Sie die Standorte der Bäume variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_5.1_A1 Lösung*.

Aufgabe 2



MyKara soll über die Welt laufen bis er vor einem Baum steht. Nach jedem Schritt soll er prüfen, ob er auf einem Blatt steht. Ist das nicht der Fall, soll er ein Blatt ablegen. Wenn er auf einem Blatt steht, soll er das Blatt aufheben.



Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_5.1_A2 Übung Kontrollstrukturen* als Vorlage.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

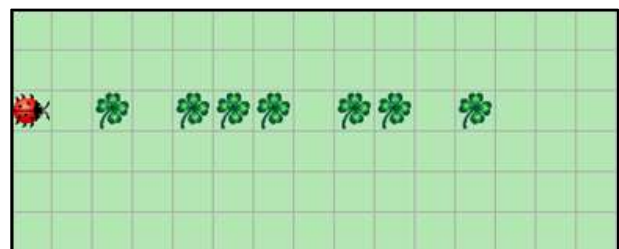
Testen Sie Ihr Ergebnis, indem Sie die Standorte der Akteure variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_5.1_A2 Lösung*.

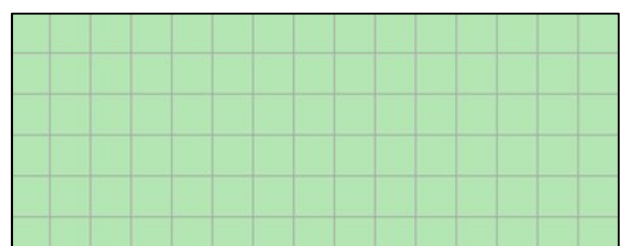
Aufgabe 3

In einer Kara-Welt wurde folgender Programmcode erfasst.

```
public void act() overrides method in Actor
{
    var int zaehler = 0
    for each (int i in 1 .. 14)
    {
        move()
        if (onLeaf())
        {
            zaehler = zaehler + 1
            removeLeaf()
        }
        turnRight()
        turnRight()
        for each (int i in 1 .. zaehler)
        {
            putLeaf()
            move()
        }
    }
}
```



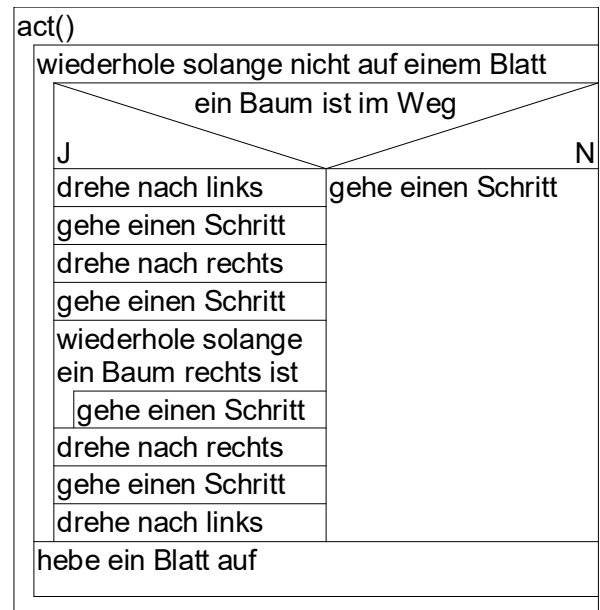
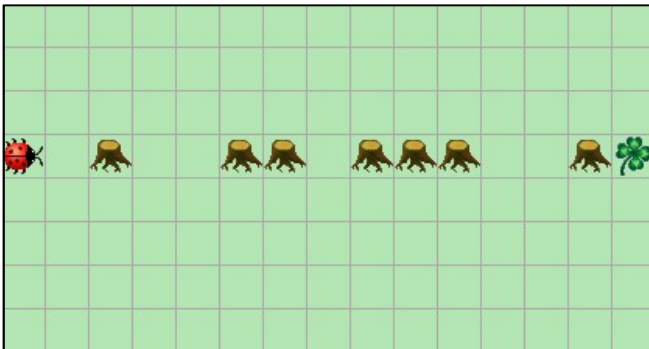
- 3.1 Beschreiben Sie die Wirkungsweise dieses Codes.
- 3.2 Skizzieren Sie das Ergebnis des Programmablaufs in der nebenstehenden Abbildung.
- 3.3 Nennen Sie den Wert, den die Variable *zaehler* nach Ablauf des Programms aufweist.



Aufgabe 4

Für die Methode *act()* einer Kara-Welt wurde folgendes Struktogramm entwickelt:

Beschreiben Sie die Wirkungsweise des daraus zu entwickelnden Programmcodes und skizzieren Sie in der folgenden Abbildung den Weg, den MyKara geht.



Aufgabe 5

Für die Methode *act()* einer Kara-Welt wurde der abgebildete Programmcode entwickelt.

Analysieren Sie den Programmablauf und erläutern Sie die syntaktischen und logischen Fehler.

```
public void act() overrides method in Actor
{
    while ( onTree() )
    {
        if ( onLeaf() )
        {
            putLeaf()
            move()
        }
        else
        {
            removeLeaf()
            move()
        }
    }
}
```

Aufgabe 6

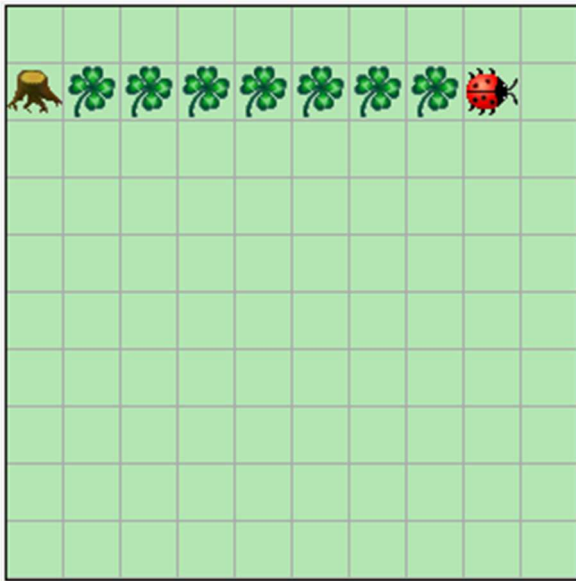
Für die Methode *act()* einer Kara-Welt wurde der abgebildete Programmcode entwickelt.

Analysieren Sie den Programmablauf und erläutern Sie die logischen Fehler.

L1_5.2 Vertiefungsaufgaben zu Kontrollstrukturen

```
public void act() overrides method in Actor
{
    while ( onLeaf() && ! onLeaf() )
    {
        move()
        if ( onLeaf() )
        {
            putLeaf()
        }
        else
        {
            removeLeaf()
        }
    }
}
```

Aufgabe 1



MyKara steht am Ende einer Blätterspur und möchte zum Baum gehen (siehe Abb.). Unterwegs sammelt er die Blätter ein. Die Blätterspur kann beliebig lang sein.

Wenn MyKara vor dem Baum steht, sollen alle Blätter aufgehoben sein.

Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario

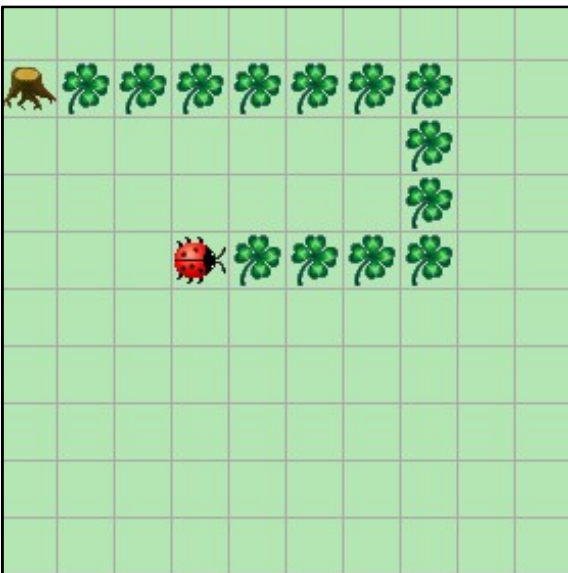
L1_5.2_A1 Vertiefungsaufgabe als Vorlage.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

Testen Sie Ihr Ergebnis, indem Sie die Standorte der Akteure variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_5.2_A1 Lösung*.

Aufgabe 2



MyKara steht am Ende der Blätterspur und möchte zum Baum gehen. Unterwegs sammelt er die Blätter ein. Die Blätterspur kann beliebig lang sein und biegt nur nach links ab.

Wenn MyKara vor dem Baum steht, sollen alle Blätter aufgehoben sein.

Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario

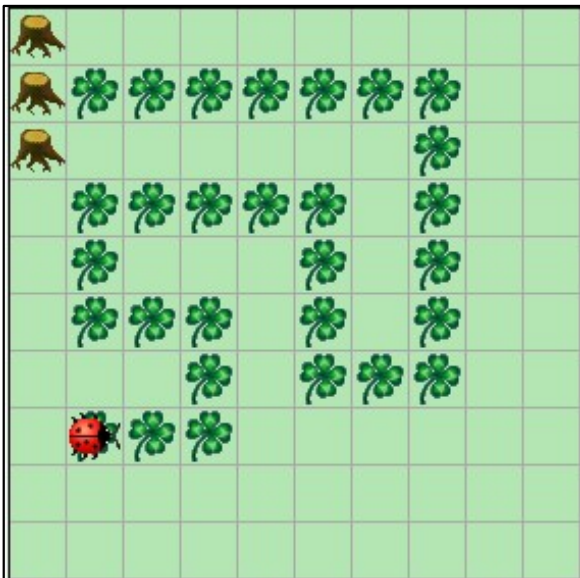
L1_5.2_A2 Vertiefungsaufgabe als Vorlage.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

Testen Sie Ihr Ergebnis, indem Sie die Standorte der Akteure variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_5.2_A2 Lösung*.

Aufgabe 3



MyKara spielt Pacman:

In der Startposition steht MyKara vor dem ersten Blatt einer langen Spur von Blättern. Die Blätterspur kann beliebig gelegt sein, verläuft aber nicht direkt nebeneinander.

Wenn MyKara vor dem Baum steht, sollen alle Blätter aufgehoben sein.

Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario

L1_5.2_A3 Vertiefungsaufgabe als Vorlage.

Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

Testen Sie Ihr Ergebnis, indem Sie die Standorte der Akteure variieren.

Speichern Sie Ihr Ergebnis unter dem Namen *L1_5.2_A3 Lösung*.