

Grundlagen der Programmierung

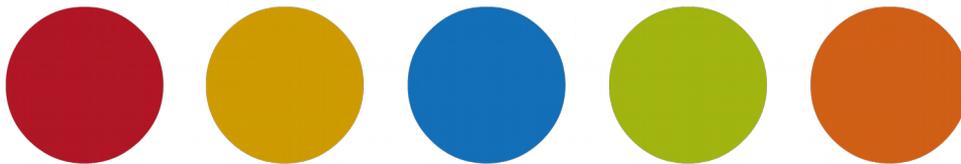
in Python

Skript

Materialsammlung

Schulung:	Informatik und Wirtschaftsinformatik
-----------	--------------------------------------

Stand: 19. Apr 2020



© Christine Janischek



Inhaltsverzeichnis

1	Sequenzen L1 1.....	3
2	Die zählergesteuerte Schleife (Zählerschleife) L1 1.2.....	7
3	Die kopfgesteuerte Schleife L1 1.3.....	14
4	Die Alternative L1 1.4.....	19
5	Vertiefung: Kontrollstrukturen L1 1.5.....	24
6	Einführung in Python – Ausgabe von Zeichenketten L2 1.....	31
7	Einführung in Python – Variablen L2 2.....	33
8	Einführung in Python – Rechenoperationen L2 3.....	36
9	Einführung in Python – Eingabe mittels input L2 4.....	40
10	Einführung in Python – Funktionen L2 5.....	47
11	Einführung in Python – Verzweigungen L3 1.....	51
12	Einführung in Python – Wiederholungen L3 2.....	62
13	Vertiefung – Kontrollstrukturen L3 3.....	68
14	Einführung in Python – Benutzeroberflächen (GUI) L3 4.....	78



1 Sequenzen L1 1

Sequenzen L1 1

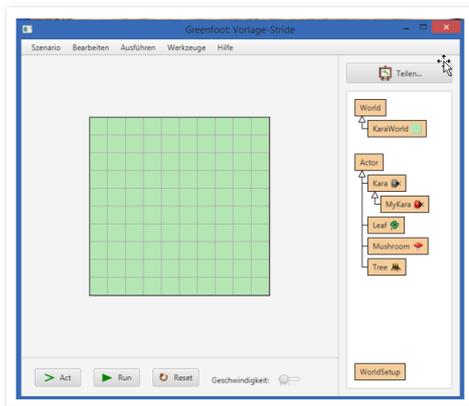


Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L1 1.1 Die Sequenz
---------------	---

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.

Thema:	Grundlagen der Programmierung in Python – Übungsaufgabe <small>Quelle: L1 1.2 Die Sequenz</small>
---------------	---

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:

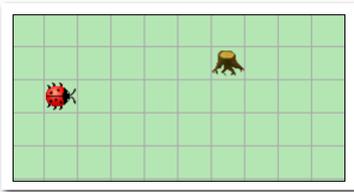


Entwickeln Sie eine Lösung, die MyKara zum Kleeblatt führt, indem er immer wieder den Weg zwischen den Bäumen wählt (siehe Abb.). Beim Kleeblatt angekommen, soll er dieses aufheben.

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.

2 Die zählergesteuerte Schleife (Zählerschleife) L1 1.2

Die zählergesteuerte Schleife

(Zählerschleife) L1 1.2



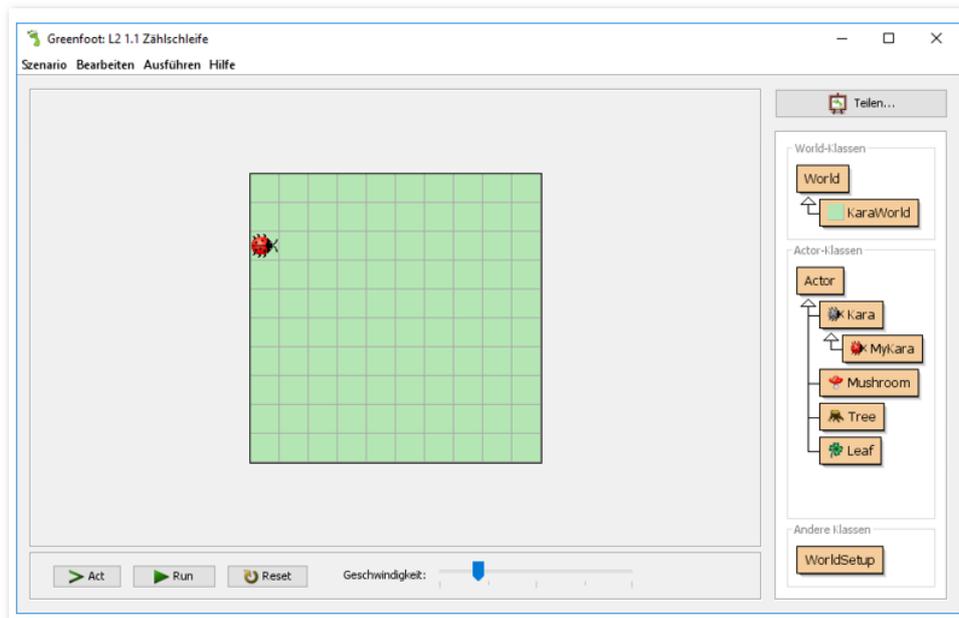
Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L1 1.2.1 Die Zählerschleife - For Schleife
--------	---

(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:



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.

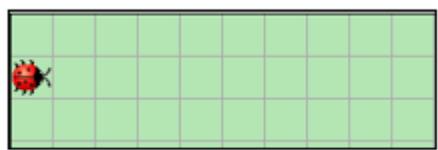
Merke: Die For-Schleife ist eine kopfgesteuerte Zählerschleife!

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

Thema:	Grundlagen der Programmierung in Python – Übungsaufgaben Teil 1 Quelle: L1 1.2.2 For Schleife
--------	---

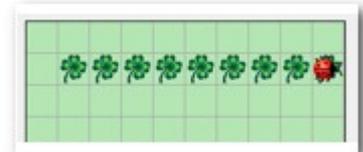
Öffnen Sie zur Bearbeitung der folgenden Aufgabenstellungen jeweils das Szenario "Vorlage-Stride".

Aufgabe 1

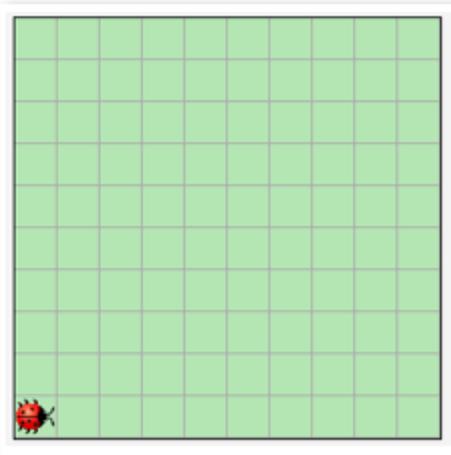


MyKara macht neun Schritte und legt dabei je-des Mal ein Blatt ab, so dass eine Art „Blattweg“ vom linken zum rechten Spielfeldrand entsteht.

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.

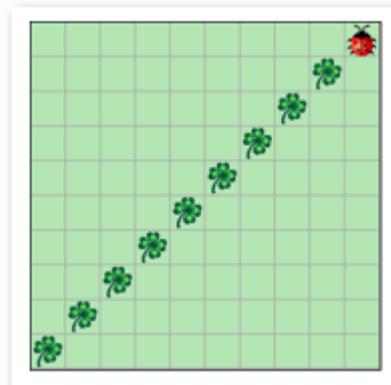


Aufgabe 2

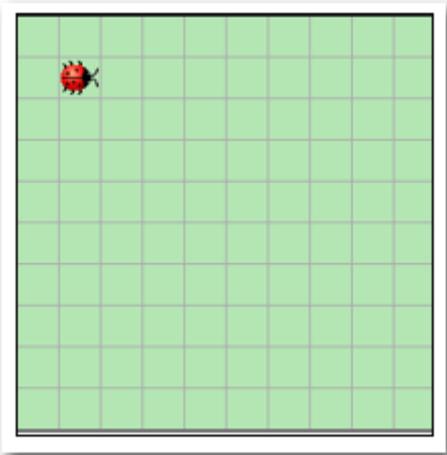


MyKara soll auf dem Spielfeld die Diagonale von links unten nach rechts oben mit Blättern auslegen. In der Startposition steht MyKara in der unteren linken Ecke.

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.

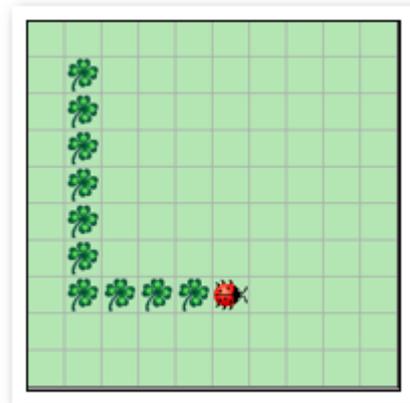


Aufgabe 3



MyKara soll mit Blättern ein "L" schreiben, wo-bei 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()
    }
}
```

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.

Thema:	Grundlagen der Programmierung in Python – Übungsaufgaben Teil 2 <small>Quelle: L1 1.2.3 For Schleife</small>
--------	--

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 "zaehler"	Nr. des Durchlaufs	Ausgabe auf Bildschirm

Aufgabe 3

```
act()
  Zähle i von 0 bis 10, Schrittweite 2
  Ausgabe: i
```

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

Aufgabe 4

```
act()
Zähle i von 0 bis 10, Schrittweite 4
Ausgabe: i
```

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

3 Die kopfgesteuerte Schleife L1 1.3

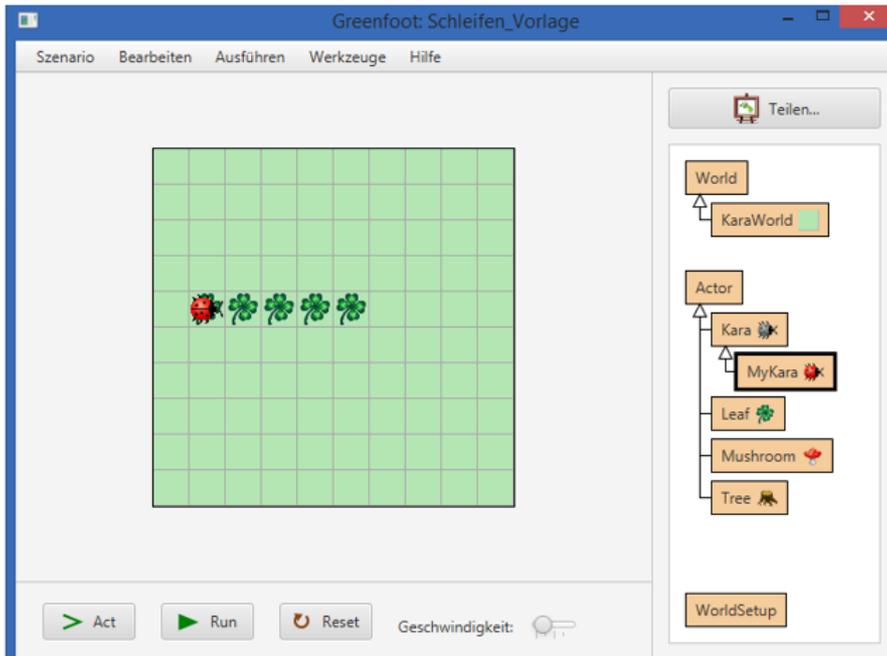
Die kopfgesteuerte Schleife L1 1.3



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L1 1.3.1 While-Schleife
--------	--

(I) Problemstellung

Öffnen Sie das Szenario "Vorlage-Stride" aus dem Ordner Vorlagen_Szenarien und platzieren Sie ei-nen 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?

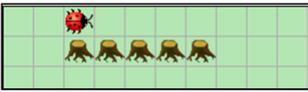
2 Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen L1_3.1 While-Schleife.

Merke: Die klassische While-Schleife ist ebenfalls eine kopfgesteuerte Schleife!

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

Thema:	Grundlagen der Programmierung in Python – Übungsaufgaben Quelle: L1 1.3.2 While-Schleife
--------	--

Aufgabe 1



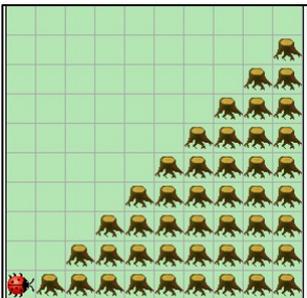
MyKara soll den Bäumen entlang gehen, bis rechts von ihm kein Baum mehr ist.

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*.



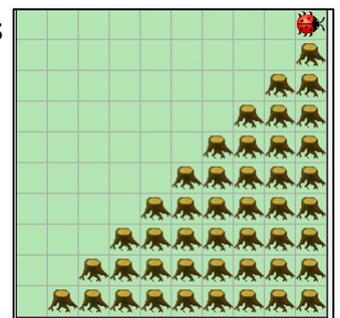
Aufgabe 2



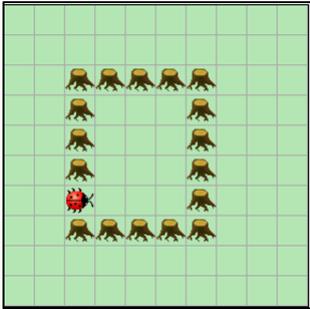
MyKara soll eine beliebig lange Treppe hochlaufen.

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*.



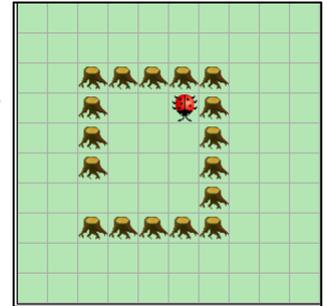
Aufgabe 3



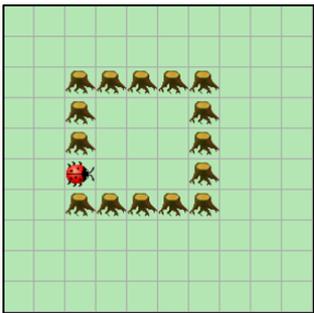
MyKara steht im Eingang seines rechteckigen Baus. MyKara soll in die obere hintere Ecke des Baus laufen (siehe Abb.).
Beachten Sie, dass der Karabau verschiedene Ausmaße haben kann.

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.



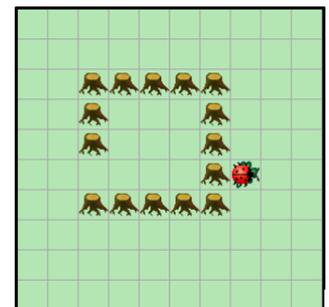
Aufgabe 4



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.

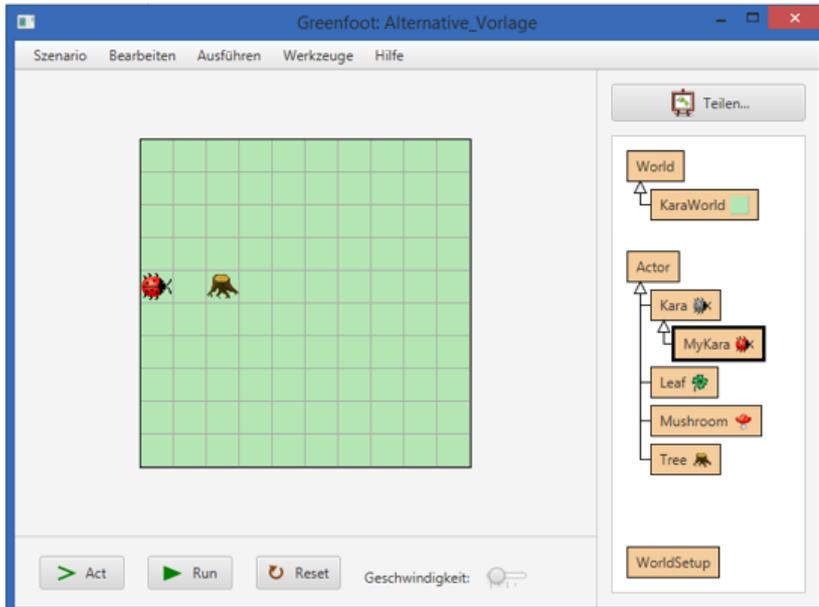
1. Erläutern Sie den logischen Fehler, den dieser Programmcode enthält.
2. Beschreiben Sie die Wirkungsweise des Programmcodes, nachdem der Fehler korrigiert wurde.

4 Die Alternative L1 1.4

Die Alternative L1 1.4



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L1 1.4.1 Alternative
--------	---



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?

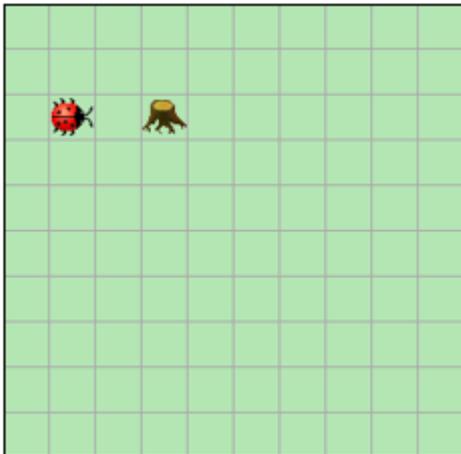
2 Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung. Speichern Sie Ihr Ergebnis unter dem Namen *L1_4.1 Alternative*.

Merke: Die Alternative ist keine Schleife sondern eine Fallunterscheidung!

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

Thema:	Grundlagen der Programmierung in Python – Übungsaufgaben Quelle: L1 1.4.2 Alternative
--------	---

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.

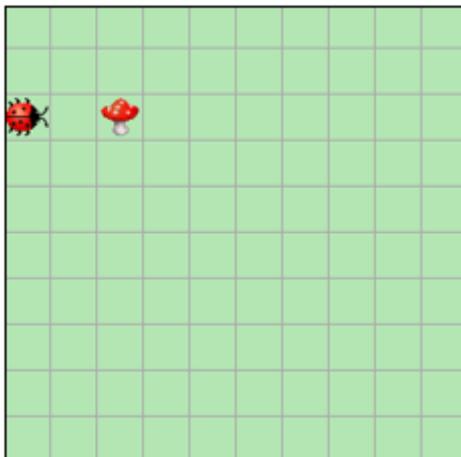
Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.



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

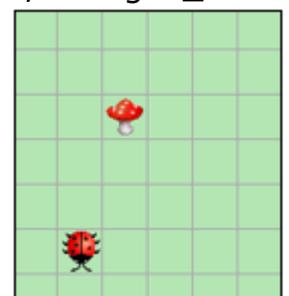


MyKara hat Angst vor Pilzen (Mushroom). Nachdem er einen Schritt vorwärts gemacht hat, soll er prüfen, ob vor ihm ein Pilz steht. Ist dies der Fall, soll er sich nach rechts drehen und drei Schritte vorwärts gehen. Ist dies nicht der Fall, soll er nur einen Schritt vorwärts gehen.

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

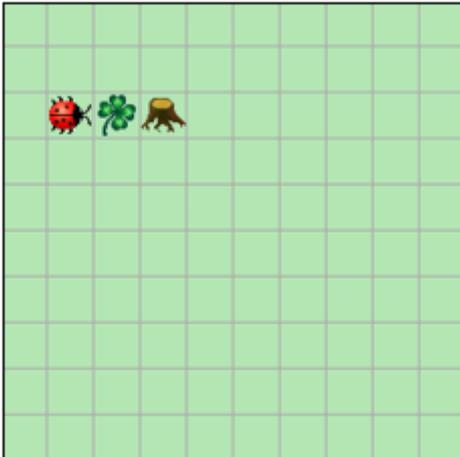
kodieren Sie die Lösung.



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.

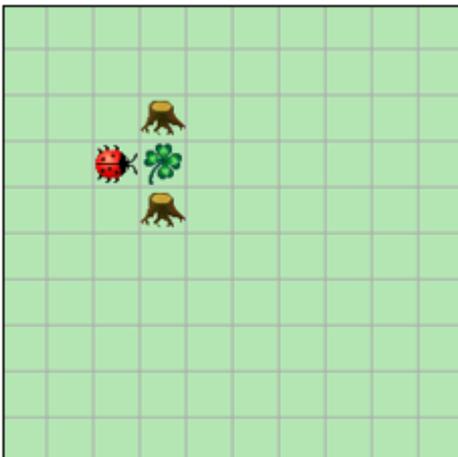


Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

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.

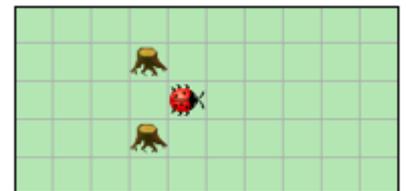
Befinden sich neben MyKara keine zwei Bäume, soll er drei Schritte vorwärts gehen.

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.

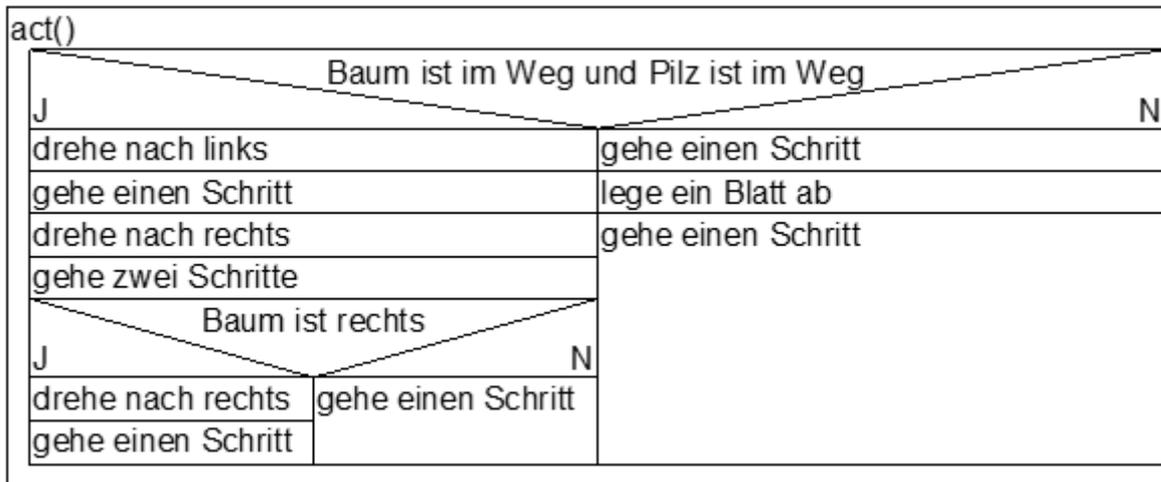
Testen Sie Ihr Ergebnis, indem Sie die Standorte der Akteure variieren.

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:

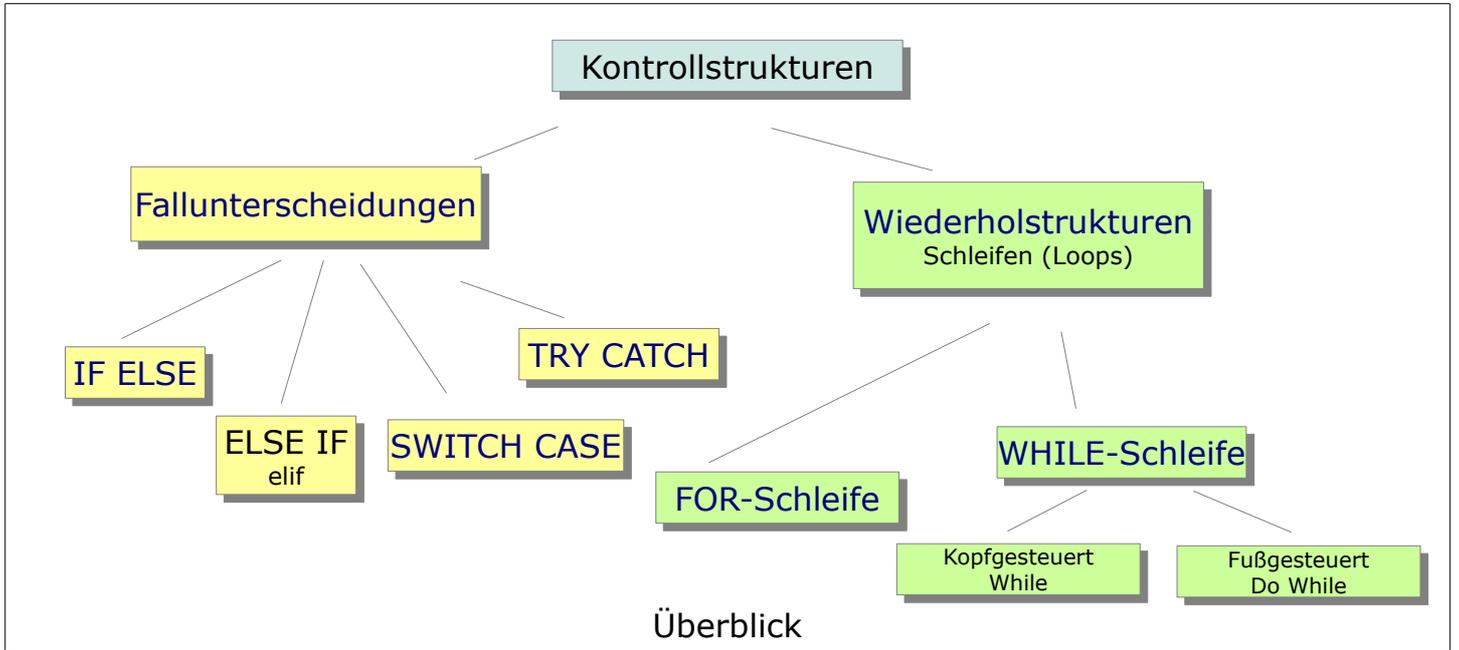


Analysieren Sie das Struktogramm und erläutern Sie die zwei logischen Fehler.

5 Vertiefung: Kontrollstrukturen L1 1.5

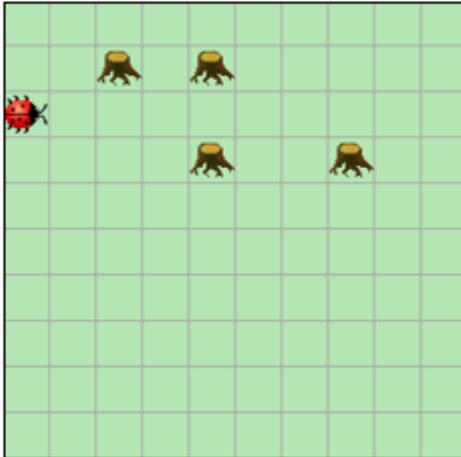
Vertiefung: Kontrollstrukturen L1 1.5





Thema:	Grundlagen der Programmierung in Python – Übungsaufgaben Quelle: L1 1.5.1 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.

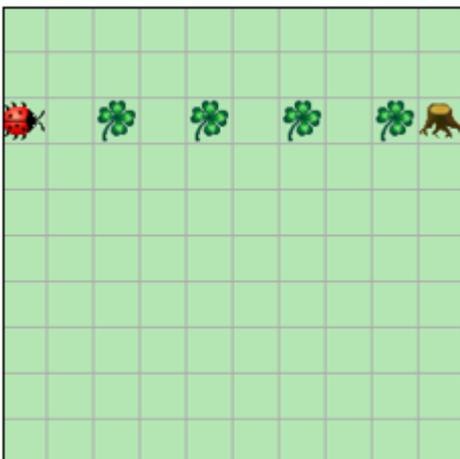


Erstellen Sie ein Struktogramm zur Lösung des beschriebenen Problems und kodieren Sie die Lösung.

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

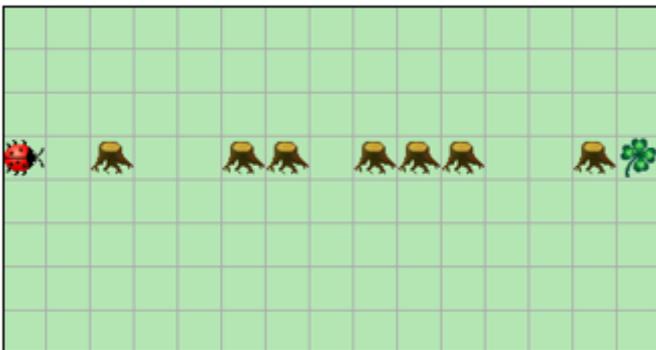
In einer Kara-Welt wurde folgender Programmcode erfasst.

```
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()
    }
}
```



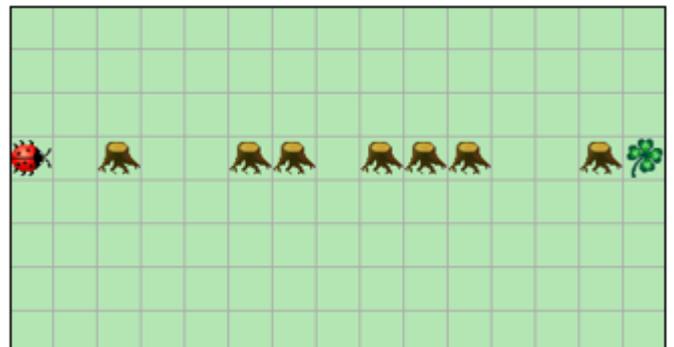
1. Beschreiben Sie die Wirkungsweise dieses Codes.
2. Skizzieren Sie das Ergebnis des Programmablaufs in der nebenstehenden Abbildung.
3. Nennen Sie den Wert, den die Variable *zaehler* nach Ablauf des Programms aufweist.

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

```

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

```

Für die Methode `act()` einer Kara-Welt wurde der abgebildete Programmcode entwickelt. Analysieren Sie den Programmablauf und erläutern Sie die syntaktischen und logischen Fehler.

Aufgabe 6

Für die Methode `act()` einer Kara-Welt wurde der abgebildete Programmcode entwickelt. Analysieren Sie den Programmablauf und erläutern Sie die logischen Fehler.

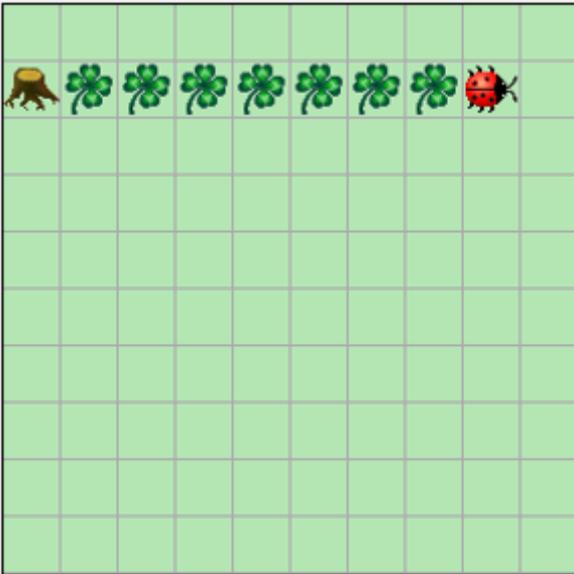
```

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

```

Thema:	Grundlagen der Programmierung in Python – Vertiefungsaufgaben Quelle: L1 1.3.2 Kontrollstrukturen
--------	---

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.

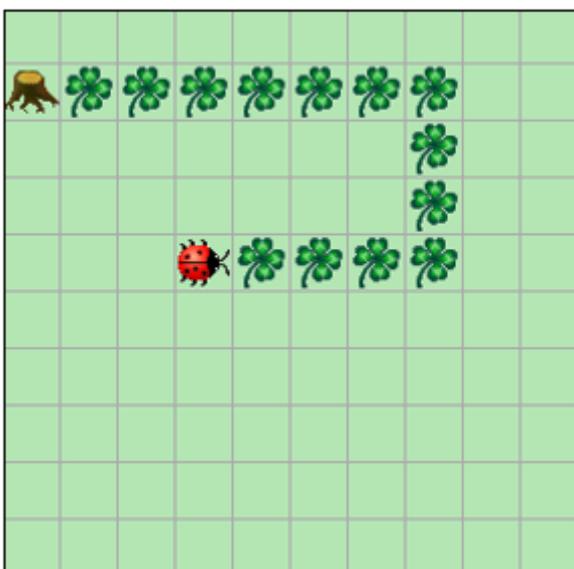
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_5.2_A1 Vertiefungsaufgabe* als Vorlage.

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.

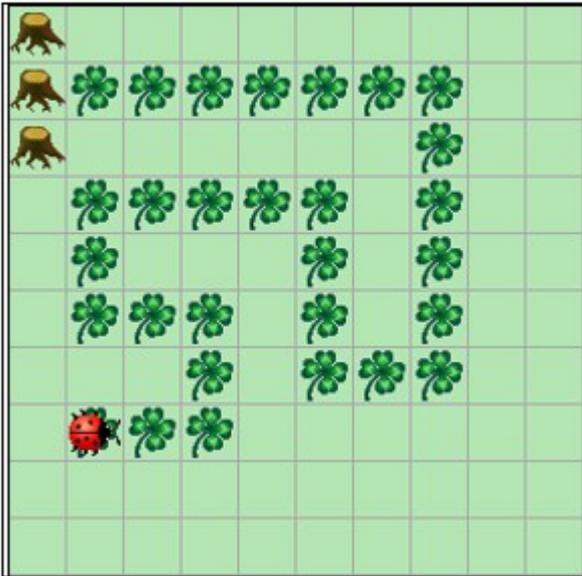
Verwenden Sie aus dem Ordner Aufgaben/Vorlagen_Szenarien das Szenario *L1_5.2_A2 Vertiefungsaufgabe* als Vorlage.

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*.

6 Einführung in Python – Ausgabe von Zeichenketten L2 1

E
nführung
n Python Ausgab

L2 1



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L2 1 Ausgabe von Zeichenketten
--------	---

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial L2_1 Information_Ausgabe_von_Daten sowie das Lernvideo <https://vimeo.com/332928990>, in dem erklärt wird, wie man ein Python-Programm mit einer einfachen Textausgabe schreibt.

(I) Problemstellung

Schreiben Sie ein Python-Programm, das Ihre persönlichen Daten auf dem Bildschirm ausgibt. Speichern Sie dieses Programm als L2_1_Textausgabe.py ab.

(II) Problemanalyse

(1) So soll die Bildschirmausgabe des Programms aussehen:

(III) Struktogramm

(IV) Programmcode (Python-Code)

7 Einführung in Python – Variablen L2 2

E

nführung

n Python *Varia-*
bl

L2 1



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag <small>Quelle: L2 2 Variablen - Kurswahl</small>
---------------	---

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informations-material L2_2_1 Information_Variablen_Zeichenkette, L2_2_2 Information_Variablen_Zahlen und L2_2_3 Information_Variablen_Wahrheitswert.

(I) Problemstellung

Schreiben Sie ein Programm, das die Daten eines Schülers in mehreren Variablen speichert und hinterher ausgibt. Zu speichern sind: Name, Vorname, Alter, Notendurchschnitt, zweite Fremdsprache(True/False) und Musik (True/False). Diese Daten sollen dann tabellarisch im Browser ausgegeben werden.

(II) Problemanalyse

1. Welche Ausgabedaten will man erhalten?
2. Welche Eingabedaten werden zur Bearbeitung benötigt?
3. Welche Eigenschaften haben Ausgabedaten? (Variablenliste)

Bedeutung	Inhalt	Variablenname

4. So soll die Bildschirmausgabe des Programms aussehen:

(III) Struktogramm

Verändert den Quellcode zur Übung so, dass die folgende Ausgabe erzeugt wird:

Deklaration und Initialisierung: nachname als zeichenkette = "Mustermann"
Deklaration und Initialisierung: vorname als zeichenkette = "Max"
Deklaration und Initialisierung: alter als Ganzzahl = "Max"
Deklaration und Initialisierung: schnitt als Dezimalzahl = 1.8
Deklaration und Initialisierung: zweit_fremd als Wahrheitswert = True
Deklaration und Initialisierung: musik als Wahrheitswert = False
Ausgabe: "name:", vorname, nachname
Ausgabe: "Alter:", alter
Ausgabe: "Notenschnitt:", schnitt
Ausgabe: "2. Fremdsprache belegt:", zweit_fremd
Ausgabe: "Musik belegt:", musik

(IV) Programmcode (Python-Code)

8 Einführung in Python – Rechenoperationen L2 3

i
i
e
r
e
n

E
nführung
n Python *Re-*
ch
nope-
atio-
n
L2 3

Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L2_3 Rechenoperationen - Taschenrechner
---------------	--

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial L2_3 Information Rechenoperatoren.

(I) Problemstellung

Entwerfen Sie ein Programm, das die Rechenoperationen plus, minus, mal, geteilt und hoch durchführt. Definieren Sie dazu zwei Variablen, die Sie für die Berechnungen nutzen und geben Sie das Ergebnis am Bildschirm aus.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Daten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben Ausgabedaten? (Variablenliste)

Bedeutung	Inhalt	Variablenname

(4) So soll die Bildschirmausgabe des Programms (erstes Teilergebnis) aussehen:

(5) Variieren Sie den Quellcode (zweites Teilergebnis) so, dass die Ausgabe wie folgt aussieht:

```
Zahl 1: 12 Zahl 2: 3
-----
Addition: 15
Subtraktion: 9
Multiplikation: 36
Division: 4.0
Potenz: 1728
```

(III) Struktogramm

(IV) Programmcode (Python-Code)



9 Einführung in Python – Eingabe mittels input L2 4

E *i* *n Python* Eingab *nführung*

L2 4



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L2 1 4.1 Eingabe input Kreis
---------------	---

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial L2_4_1 Information_Eingabe_input_string und L2_4_2 Information_Eingabe_input_zahlen.

(I) Problemstellung

Bei den bisherigen Programmen waren alle Werte fest vom Programmcode vorgegeben. Es bestand keine Möglichkeit für die Programmbeutzer, selber Werte in das Programm einzugeben. Diese Option wird jetzt gezeigt. Schreiben Sie dazu ein Programm, das den Inhalt einer Kreisfläche berechnet. Der Benutzer soll den Radius im Programm auf Anforderung eingeben. Das Programm errechnet die passende Kreisfläche und gibt das Ergebnis aus.

Formel: Kreisfläche = Radius*Radius*pi

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben Eingabedaten und Ausgabedaten? (Variablenliste)

Bedeutung	Datentyp	Variable

(4) So soll die Bildschirmausgabe des Programms aussehen:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Übungsaufgabe <small>Quelle: L2 4.2 Übung: Eingabe mittels input() – Rechteck</small>
---------------	---

Hinweis: Beachten Sie bei Bedarf zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial L2_4_1 Information_Eingabe_input_string und L2_4_2 Information_Eingabe_input_zahlen.

(I) Problemstellung

Schreiben Sie ein kleines Programm, das den Umfang und den Flächeninhalt eines Rechtecks berechnet. Der Benutzer gibt dazu die Länge und die Breite des Rechtecks ein. Das Programm berechnet die Ergebnisse und gibt diese aus.

(II) Problemanalyse

- (1) Welche Ausgabedaten will man erhalten?
- (2) Welche Eingabedaten werden zur Bearbeitung benötigt?
- (3) Welche Eigenschaften haben Eingabedaten und Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

- (4) So soll die Bildschirmausgabe des Programms aussehen:

- (5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Übungsaufgabe Quelle: L2 4.3 Übungsaufgabe: Berechnung Höhenmeter
---------------	---

(I) Problemstellung

Die Siedetemperatur von Wasser beträgt auf Höhe des Meeresspiegels 100o Celsius. Bei zunehmender Höhe verringert sich die Siedetemperatur um 1o Celsius pro 300 Höhenmeter.

Entwerfen Sie ein Programm, mit dem man mithilfe der Siedetemperatur des Wassers die aktuelle Höhe bestimmen kann.

Die Eingabe der Daten soll am Bildschirm erfolgen.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben Eingabedaten und Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) So soll das Programm aussehen:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



10 Einführung in Python – Funktionen L2 5

i
i
e
n

E
nführung
n Python Funktio-
n
L2 5



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L2 5.1 Funktion Urlaubsfahrt
---------------	---

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial L2_5_1 Information_Funktion_ohne_Parameter.docx, L2_5_2 Informati-on_Funktion_mit_Parameter.docx und L2_5_3 Informati-on_Funktion_mit_Rückgabewert.docx

(I) Problemstellung

Sie möchten in den Ferien selbst mit dem Auto in den Urlaub fahren. Schreiben Sie ein Programm, das Ihnen die Benzinkosten für Ihre Urlaubsfahrt berechnet.

Da die Berechnung der gesamten Benzinkosten etwas länger bzw. komplexer ist als die bisherigen Berechnungen, soll diese Berechnung in eine eigene Funktion ausgelagert werden. Wie das gemacht wird, erfahren Sie in den oben angegebenen Informationsmaterialien. Bearbeiten Sie alle drei angegebenen Materialien (mitsamt der kleinen Herausforderungen am Ende der Informationsblätter) und benutzen Sie anschließend für dieses Programm eine Funktion mit Parameter und Rückgabewert.

Folgende Parameter soll diese Funktion erhalten: Benzinpreis, Verbrauch Auto, Länge der Strecke.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben Eingabedaten und Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) So soll die Bildschirmausgabe des Programms aussehen:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



11 Einführung in Python – Verzweigungen L3 1

E

nführung

n Python Verzweigun-
g

L3 1

Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 1.1 Einseitige Verzweigung - Abiball
---------------	--

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial L3_1_1_Information_Verzweigungen_einseitig.docx

(I) Problemstellung

In diesem Projekt soll überprüft werden, welcher Betrag zu zahlen ist, wenn eine Karte für den Abiball 30,00 Euro kostet und für Bestellungen von drei oder mehr Karten ein Rabatt von 20% nachgelassen wird.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben Eingabedaten und Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 1.2 Zweiseitige Verzweigung Preisvorschlag
---------------	--

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial

L3_1_2_Information_Verzweigungen_zweiseitig.docx

(I) Problemstellung

In diesem Projekt können Gebote für einen zum Kauf angebotenen Artikel abgegeben werden. In dem Programmcode ist der Mindestpreis (hier 24.99) hinterlegt. Ist der von dem Anwender vorgeschlagene Kaufpreis geringer, wird der Text „Ihr Angebotspreis liegt unterhalb des Mindestpreises!“ ausgegeben. Andernfalls erscheint der Text „Herzlichen Glückwunsch, Sie haben den Artikel erworben!“.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben Eingabedaten und Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms (falls Angebotspreis niedriger als 24.99 ist):

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 1.3 Geschachtelte Verzweigung Mietzuschuss
---------------	--

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial

L3_1_3_Information_Verzweigungen_geschachtelt.docx.

(I) Problemstellung

Schreiben Sie ein Programm, das den Mietzuschuss in Abhängigkeit vom Mietpreis berechnet.

Bei einer Miete von weniger als 500,00 Euro beträgt der Zuschuss 2%. Von 500,00 Euro bis unter 1000,00 Euro beträgt der Zuschuss 5% und ab 1000,00 Euro erhält man einen Zuschuss von 7%.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben Eingabedaten und Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 1.5 Verzweigung und Logische Operatoren Eintritt
---------------	--

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informations-material

L3_1_4_Information_Verknüpfte_Bedingungen.docx.

(I) Problemstellung

In diesem Projekt soll überprüft werden, ob ein Besucher eines Tierparks ein Kind (bis 12 Jahre) oder ein Senior (ab 60 Jahre) ist. Kinder und Senioren zahlen 8,00 Euro Eintritt, für alle anderen beträgt der Eintrittspreis 12,00 Euro.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben Eingabedaten und Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 1.6 Verzweigung und logische Operatoren Autovermietung
---------------	--

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial

L3_1_4_Information_Verknüpfte_Bedingungen.docx.

(I) Problemstellung

Die Firma Rent A Car in Stuttgart vermietet Oldtimer für 50,00 € pro Tag. Alle Kunden, die nicht in Stuttgart wohnen, erhalten einen Sonderrabatt von 10%.

Das Projekt soll nach Eingabe des Wohnorts und der Mietdauer die Mietkosten ermitteln und anzeigen.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben die Eingabedaten und die Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



12 Einführung in Python – Wiederholungen L3 2

Einführung in Python Wiederholung

g

e
n

L3 2



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 2.1 Wiederholungen mit For-Schleife Taschengeld
---------------	---

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial

L3_2_1 Information_for_Schleife.docx

(I) Problemstellung

Schreiben Sie ein Programm, das den Betrag des Taschengelds in Abhängigkeit vom Alter ausgibt. Das Taschengeld wird vom 6. bis zum 21. Lebensjahr ausgezahlt. Dabei sind der Anfangsbetrag und die jährliche Erhöhung vom Benutzer einzugeben.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben die Eingabedaten und die Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag <small>Quelle: L3 2.2. Wiederholung mit While-Schleife Fischteich</small>
---------------	--

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial

L3_2_2 Information_while_Schleife.docx

(I) Problemstellung

Der Fischbestand eines Teichs beträgt zu Beginn eines Jahres drei Fische. Jedes Jahr verdoppelt sich der Fischbestand.

Erstellen Sie ein Programm, das die Dauer in Jahren ermittelt, bis ein bestimmter, vom Anwender einzugebender Fischbestand erreicht bzw. überschritten wird.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben die Eingabedaten und die Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



13 Vertiefung – Kontrollstrukturen L3 3

Vertiefung Kontrollstruktu- r

L3 3

e
n



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 3.2 Vertiefung For-Schleife Finanzanlage
---------------	--

(I) Problemstellung

In diesem Projekt soll überprüft werden, welchen Zielwert eine Finanzanlage, die zu einem bestimmten Zinssatz für eine bestimmte Anlagedauer (in Jahren) durch Zinseszinsvermehrung erreicht. Für jedes Jahr, die die Finanzanlage läuft, soll der Wert am Ende des Jahres ausgegeben werden.

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben die Eingabedaten und die Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 3.2 While-Schleife Zielwert Verzinsung
---------------	--

(I) Problemstellung

Auf einem Sparbuch wird ein Geldbetrag angelegt. Für diesen Betrag gibt es einen bestimmten Zinssatz. In diesem Projekt soll überprüft werden, wie viele Jahre es dauert bis dieser Betrag einen bestimmten Zielwert erreicht oder überschreitet. Dabei sollen die Jahresendwerte der Finanzanlage jeweils ausgegeben werden. (Zinseszinsen nicht vergessen!)

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben die Eingabedaten und die Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 3.3. Vertiefung Geschachtelte Verzweigung Anmeldung BG
---------------	--

(I) Problemstellung

In diesem Projekt soll überprüft werden, ob eine Anmeldung mit einem mittleren Bildungsabschluss an einem beruflichen Gymnasium in Baden-Württemberg möglich ist oder nicht. Die Bedingungen dafür lauten wie folgt:

Abschlusszeugnis (Mittlerer Bildungsabschluss) der Realschule, der zweijährigen Berufsfachschule, der Berufsaufbauschule oder der Werkrealschule mit einem Notendurchschnitt von mindestens 3,0 aus Deutsch, Mathematik und Englisch (kein Fach schlechter als Note ausreichend).

(II) Problemanalyse

(1) Welche Ausgabedaten will man erhalten?

(2) Welche Eingabedaten werden zur Bearbeitung benötigt?

(3) Welche Eigenschaften haben die Eingabedaten und die Ausgabedaten? (Variablenliste)

Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:

(5) Verarbeitung

(III) Struktogramm

(IV) Programmcode (Python-Code)



Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: L3 3.4 Vertiefung Verzweigung und logische Operatoren Anmeldung BG
---------------	---

(I) Problemstellung

In diesem Projekt soll der gleiche Sachverhalt wie in L3_3_3 umgesetzt werden. Hier sollen jedoch logische Operatoren anstelle einer Verschachtelung der Verzweigungen verwendet werden.

(II) Problemanalyse

- (1) Welche Ausgabedaten will man erhalten?

- (2) Welche Eingabedaten werden zur Bearbeitung benötigt?

- (3) Welche Eigenschaften haben die Eingabedaten und die Ausgabedaten? (Variablenliste)

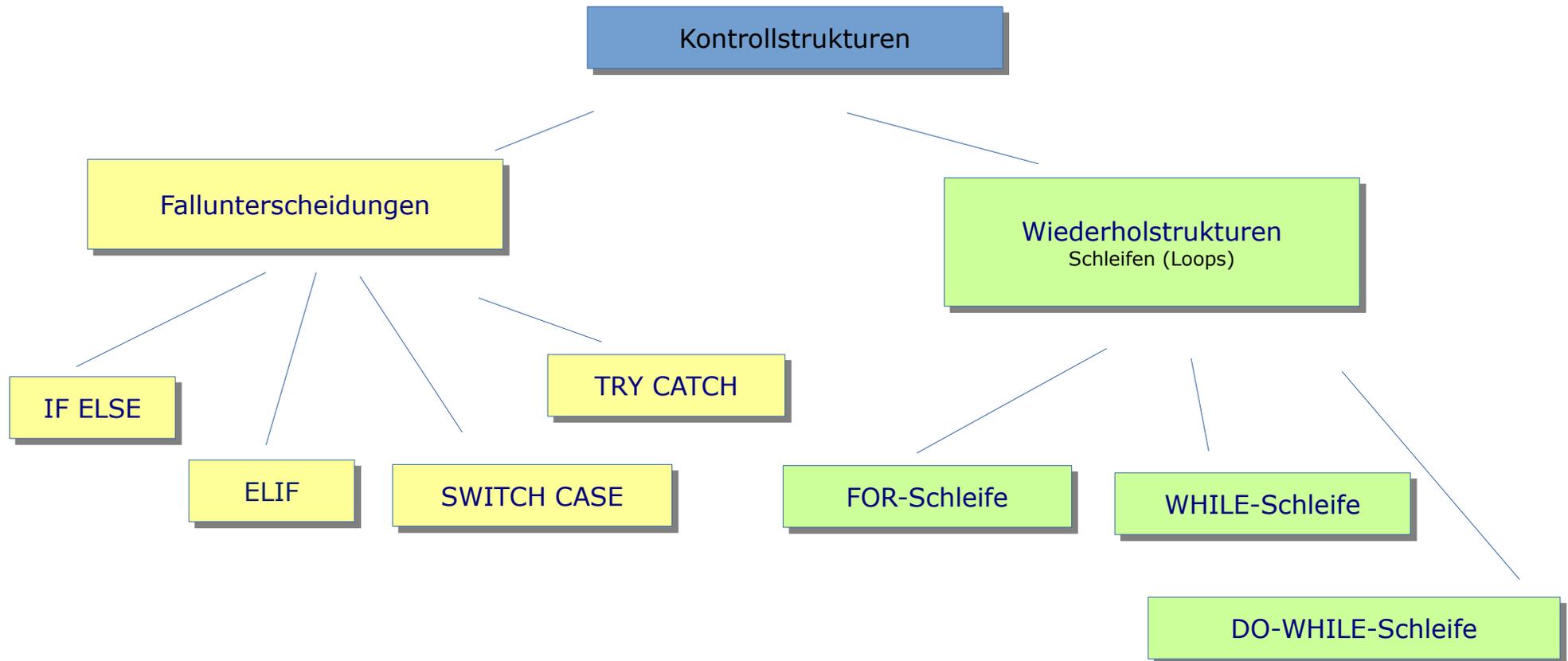
Bedeutung	Typ	Variable

(4) Bildschirmausgabe des Programms:**(5) Verarbeitung**

(III) Struktogramm

(IV) Programmcode (Python-Code)





14 Einführung in Python – Benutzeroberflächen (GUI) ^{L3 4}

Vertiefung in Python

B

e

n

L3 4

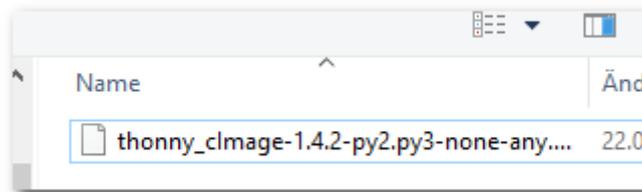


Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: Christine Janischek – Der Taschenrechner (noop)
--------	---

In diesem Projekt soll der Taschenrechner eine Benutzeroberfläche bekommen. Stellen Sie dazu sicher, dass Sie die Programmiererweiterung:

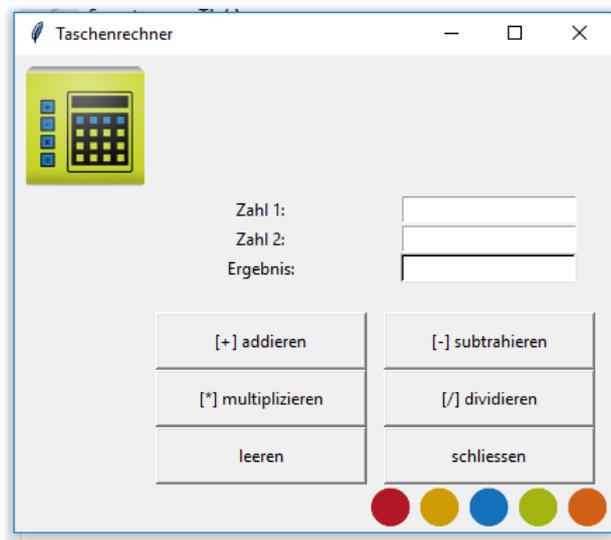
`thonny_cImage-1.4.2-py2.py3-none-any.whl`

in der Entwicklungsumgebung Thonny installiert haben. Dazu gehen Sie in der Menüleiste Thonny → Tools → Manage packages → Install from local files (here) → Die Datei



Die Erweiterung stellt sicher, dass Sie auf den Benutzeroberflächen Grafiken (wie z.B. ein Logo) einbetten können. Das Logo und Icon erhalten Sie vom Klassenlehrer und muss sich im Projektordner befinden.

Nutzen Sie das Informationsblatt und die Vorlage Beispiel.py.



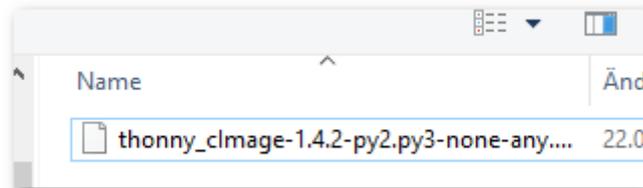
Arbeitsauftrag:

1. Erzeugen Sie ein neues Projekt um den → Taschenrechner umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus den letzten Kapiteln (→ kopieren und modifizieren), um die Aufgabe zu lösen.
2. Testen und Dokumentieren Sie alle Ergebnisse.

Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: Christine Janischek – Der Notenrechner (noop)
--------	---

In diesem Projekt soll der Notenrechner eine Benutzeroberfläche bekommen. Stellen Sie dazu sicher, dass Sie die Programmerweiterung:

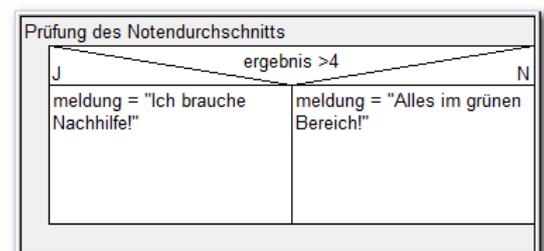
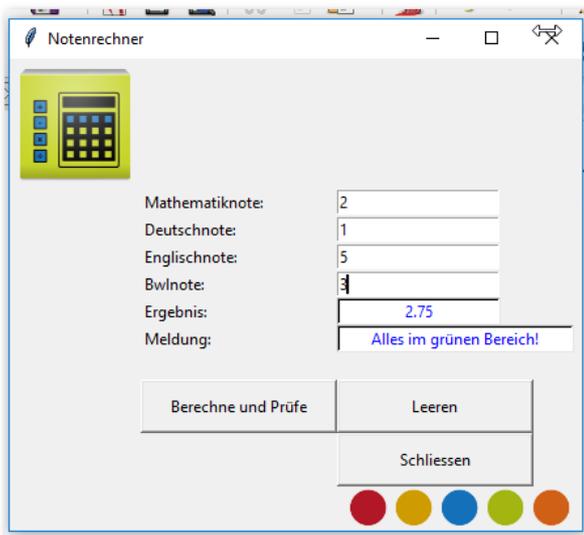
thonny_cImage-1.4.2-py2.py3-none-any.whl



in der Entwicklungsumgebung Thonny installiert haben. Dazu gehen Sie in der Menüleiste Thonny → Tools → Manage packages → Install from local files (here) → Die Datei

Die Erweiterung stellt sicher, dass Sie auf den Benutzeroberflächen Grafiken (wie z.B. ein Logo) einbetten können. Das Logo und Icon erhalten Sie vom Klassenlehrer und muss sich im Projektordner befinden.

Nutzen Sie das Informationsblatt und die Vorlage Beispiel.py.



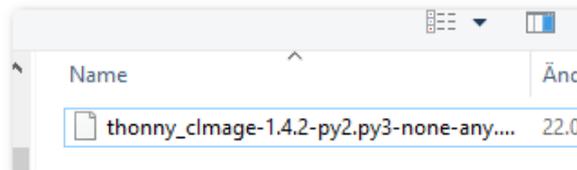
Arbeitsauftrag:

1. Erzeugen Sie ein neues Projekt um den → Notenrechner umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus den letzten Kapiteln (→ kopieren und modifizieren), um die Aufgabe zu lösen. Vorgehensweise: Lösen Sie erst die → Berechnung und erweitern Sie die Lösung dann um die → Prüfung (Hilfsfunktion/-Methode).
2. Testen und Dokumentieren Sie alle Ergebnisse.

Thema: Grundlagen der Programmierung in Python – Arbeitsauftrag
 Quelle: Christine Janischek – Der Rabattrechner (noop)

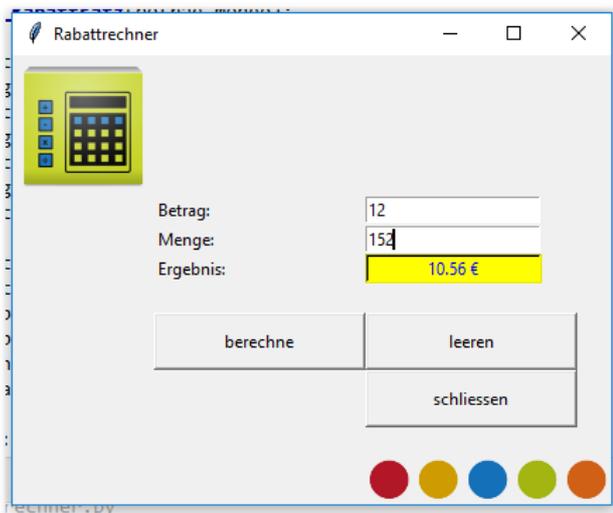
In diesem Projekt soll der Rabattrechner eine Benutzeroberfläche bekommen. Stellen Sie dazu sicher, dass Sie die Programmerweiterung:

thonny_cImage-1.4.2-py2.py3-none-any.whl



in der Entwicklungsumgebung Thonny installiert haben. Dazu gehen Sie in der Menü-Leiste Thonny → Tools → Manage packages → Install from local files (here) → Die Datei Die Erweiterung stellt sicher, dass Sie auf den Benutzeroberflächen Grafiken (wie z.B. ein Logo) einbetten können. Das Logo und Icon erhalten Sie vom Klassenlehrer und muss sich im Projektordner befinden.

Nutzen Sie das Informationsblatt und die Vorlage Beispiel.py .



Ermittle Rabattsatz				
>=150	>=100	>=50	>=20	menge?
rabattsatz = 12	rabattsatz = 10	rabattsatz = 8	rabattsatz = 6	sonst rabattsatz = 0

Für die Berechnung:

Des Rabattbetrags:
 $\text{rabattbetrag} = \text{betrag} / 100 * \text{rabattsatz};$

Des Zahlungsbetrags:
 $\text{zahlungsbetrag} = \text{betrag} - \text{rabattbetrag};$

Für die Formatierung der Ausgabe:

```
import locale
locale.format_string("%.2f", ergebnis, 1) + " €"
```

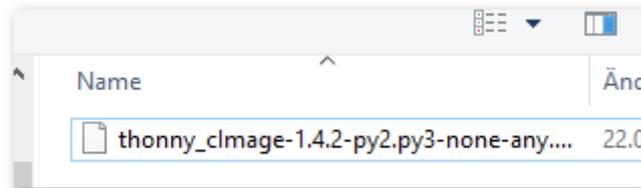
Arbeitsauftrag:

- Erzeugen Sie ein neues Projekt um den → Rabattrechner umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus den letzten Kapiteln (→ kopieren und modifizieren), um die Aufgabe zu lösen. Vorgehensweise: Lösen Sie erst die → Berechnung des Zahlungsbetrages, lagern Sie die Ermittlung des Rabattsatzes und die Ermittlung des Rabattbetrages in die Hilfsfunktionen/-Methoden → ermittle_Rabattsatz() und ermittle_Rabattbetrag() aus (Hilfsfunktion/-Methode).
- Testen und Dokumentieren Sie alle Ergebnisse.

Thema:	Grundlagen der Programmierung in Python – Arbeitsauftrag Quelle: Christine Janischek – Der Milchautomat (noop)
--------	---

In diesem Projekt soll der Milchautomat eine Benutzeroberfläche bekommen. Stellen Sie dazu sicher, dass Sie die Programmiererweiterung:

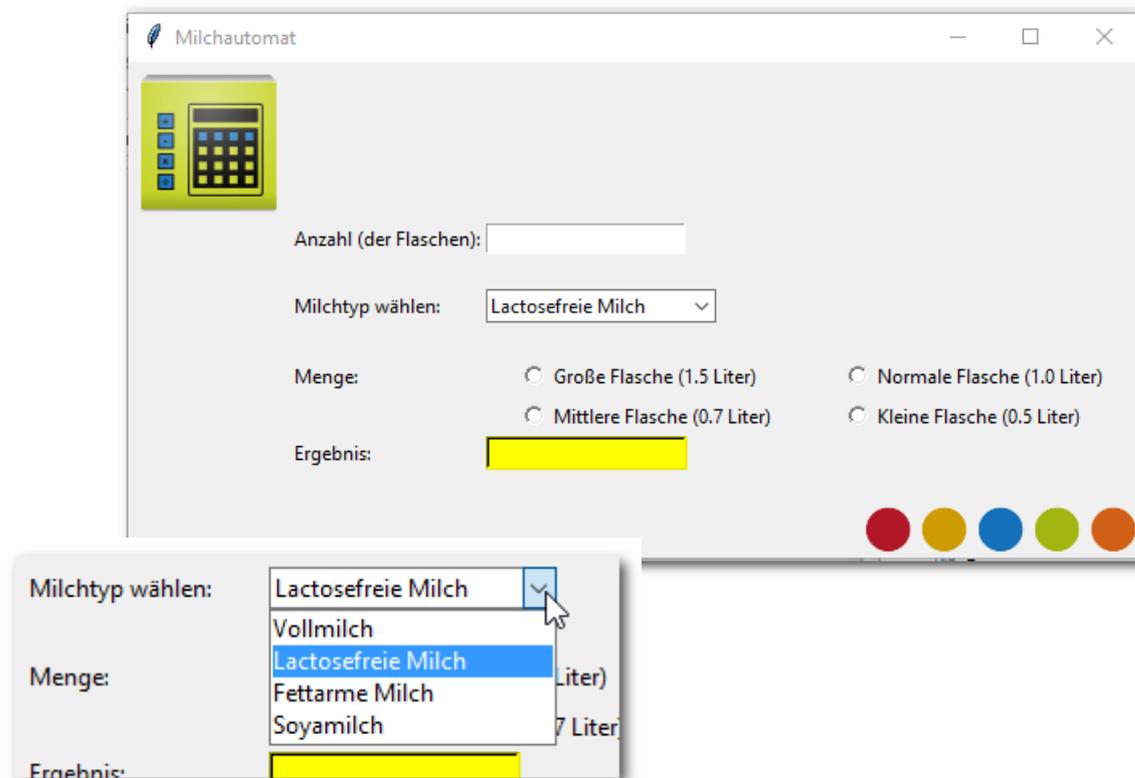
thonny_cImage-1.4.2-py2.py3-none-any.whl



in der Entwicklungsumgebung Thonny installiert haben. Dazu gehen Sie in der Menüleiste Thonny → Tools → Manage packages → Install from local files (here) → Die Datei

Die Erweiterung stellt sicher, dass Sie auf den Benutzeroberflächen Grafiken (wie z.B. ein Logo) einbetten können. Das Logo und Icon erhalten Sie vom Klassenlehrer und muss sich im Projektordner befinden.

Nutzen Sie das Informationsblatt und die Vorlage Beispiel.py .



Für die Ermittlung:

Milchpreis:

Milchtyp	Milchpreis
Vollmilch	1.30
Lactosefreie Milch	1.45
Fettarme Milch	1.10
Sojamilch	1.55

Für die Berechnung:

Flaschengröße:

value	Flaschengröße
1	1.5
2	1
3	0.7
4	0.5

Flaschenpreis:

```
flaschenpreis = milchpreis * flaschengroesse;
```

Ergebnis:

```
ergebnis = flaschenpreis * anzahl;
```

Für die Formatierung der Ausgabe:

```
import locale
```

```
locale.format_string("%.2f", ergebnis, 1) + " €"
```

Arbeitsauftrag:

1. Erzeugen Sie ein neues Projekt um den → Milchautomat umzusetzen. Nutzen Sie das Informationsmaterial und die Lösung aus den letzten Kapiteln (→ kopieren und modifizieren), um die Aufgabe zu lösen.
2. Testen und Dokumentieren Sie alle Ergebnisse.