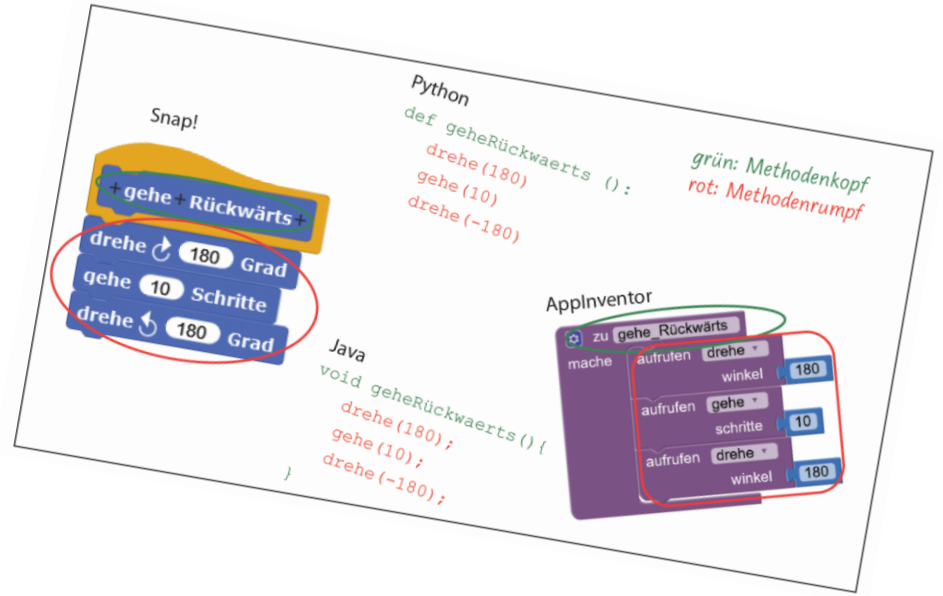
Herzlich Willkommen!

Algorithmik für die 11. Jahrgangsstufe

Franz Jetzinger

Didaktik der Informatik

Technische Universität München



Was erwartet euch?

1) Einstieg Snap!
2) Sequenz durch Algorithmik 11. Jgst. Spätbeginnend
3) Java, Python, AppInventor
4) Praxisteil

Einstieg Snap - Laufschrift

Einstieg Snap - Laufschrift

9 Laufschrift

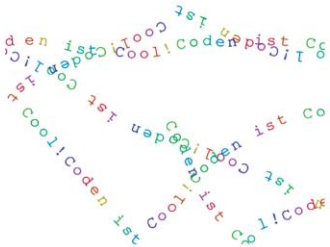
a Laufschriften werden als Blickfang im öffentlichen Raum und auf Webseiten eingesetzt. Nennen Sie ein konkretes Beispiel aus Ihrem Alltag.

b Erkunden Sie in der Vorlage die Methode *trenneNach*, mit der ein Text beispielsweise in Wörter aufgeteilt und diese in einer Liste gespeichert werden. Fassen Sie knapp Ihre Erkenntnis zusammen.

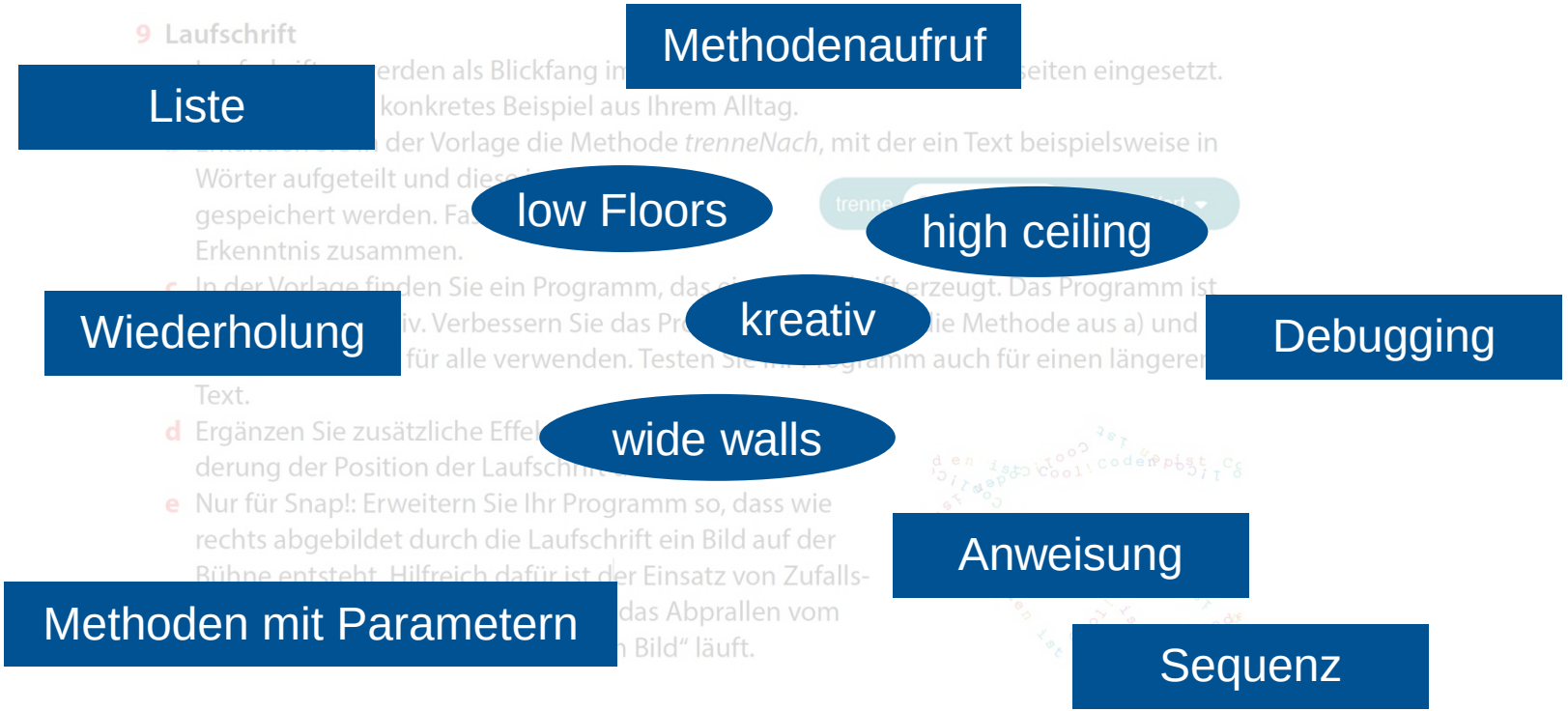
c In der Vorlage finden Sie ein Programm, das eine Laufschrift erzeugt. Das Programm ist aber sehr ineffektiv. Verbessern Sie das Programm, indem Sie die Methode aus a) und eine Wiederholung für alle verwenden. Testen Sie Ihr Programm auch für einen längeren Text.

d Ergänzen Sie zusätzliche Effekte wie Farbwechsel, Veränderung der Position der Laufschrift usw.

e Nur für Snap!: Erweitern Sie Ihr Programm so, dass wie rechts abgebildet durch die Laufschrift ein Bild auf der Bühne entsteht. Hilfreich dafür ist der Einsatz von Zufallszahlen zur Richtungsänderung und das Abprallen vom Rand, damit die Figur nicht „aus dem Bild" läuft.



Einstieg Snap – Laufschrift – informatische Konzepte



Sequenz Algorithmik 11. Jgst.

Sequenz

Programm

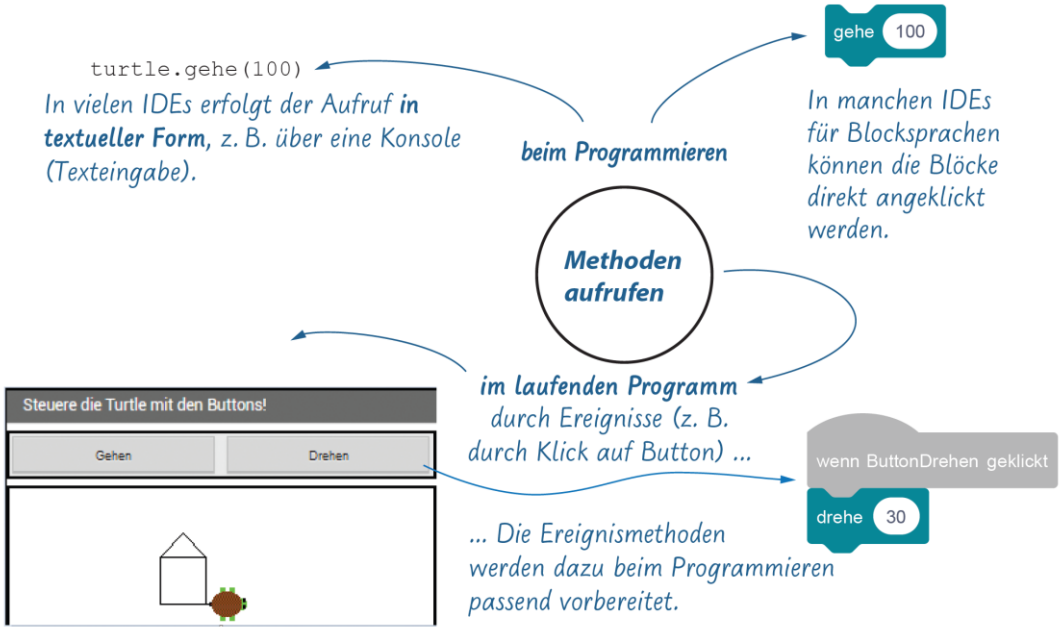
Anweisung

Programmiersprache

Methodenrump

Methodenkopf

Verständnispräzision: Methodenaufruf



Sequenz Algorithmik 11. Jgst.

1 Algorithmen

1.1 Erste Schritte: Arbeiten mit der Entwicklungsumgebung .....

1.2 Speichern von Daten: Variablen .....

Name

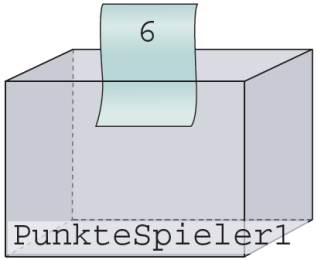
Zuweisung

Initialisierung

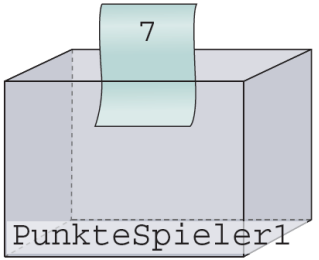
Datentyp

(Deklaration)

Fehlerquelle: Variablen und Zuweisung



setze PunkteSpieler1 auf 7



Snap!

AppInventor

Java `punkteSpieler1 = 7;`

Python `punkteSpieler1 = 7`

Sequenz Algorithmik 11. Jgst.

1 Algorithmen

1.1 Erste Schritte: Arbeiten mit der Entwicklungsumgebung .....

1.2 Speichern von Daten: Variablen .....

Qualität von Software erhöhen: Testen .....

Debugging

Pair-Programming

Fehlerkultur

Testen

Sequenz Algorithmik 11. Jgst.

1 Algorithmen

Bedingung

bedingte Anweisung

Schachtelungstiefe

Kontrollstruktur

Sequenz Algorithmik 11. Jgst.

1 Algorithmen

logische Funktion

UND

ODER

NICHT

Sequenz Algorithmik 11. Jgst.

1 Algorithmen

Wiederholung

Wiederholung mit fester Anzahl

bedingte Wiederholung

Verständnishilfe: Notional Machine

7 Ablauf von Computerprogrammen: Bedingte Anweisung mit Schablone

a Anders als ein Mensch „sieht" der Computer, welcher ein Programm ausführt, nicht den gesamten Quelltext, sondern immer nur die Zeile, die gerade ausgeführt wird. Erleben Sie zu zweit ein Programm aus der Sicht eines Computers mithilfe einer Schablone aus Papier oder Karton. In die Mitte müssen Sie ein Loch schneiden, das genau so breit und lang wie eine Zeile des folgenden Quelltextes ist.

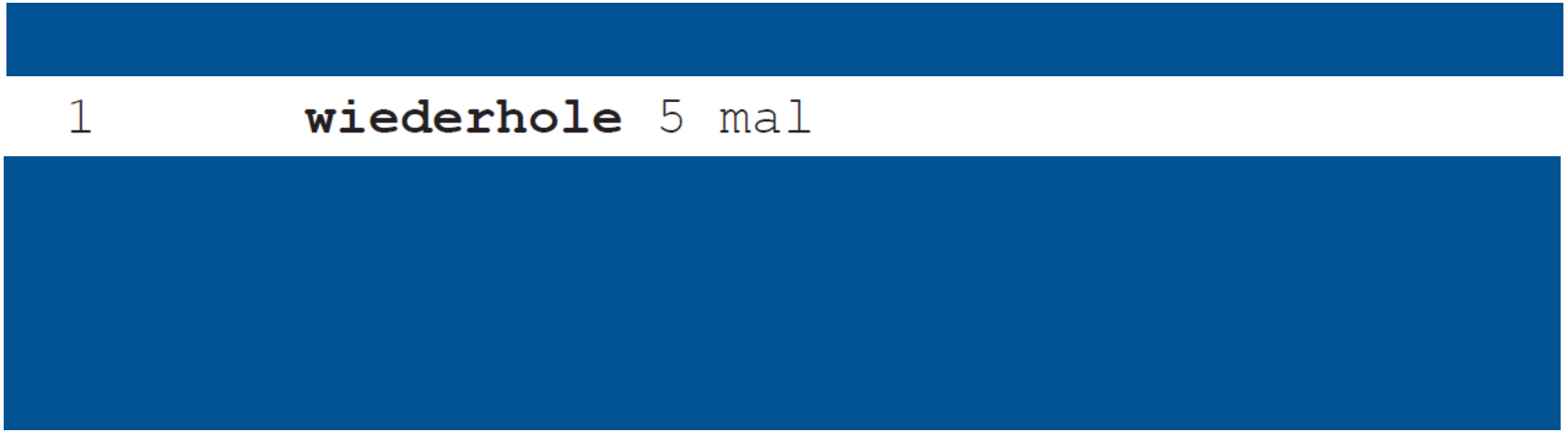
- Führen Sie damit das Programm unten aus, indem Sie das Loch zu Beginn über Zeile 1 halten und dann immer in die passende Zeile weiterschieben.
- Stellen Sie sich genauso „dumm" wie ein Computer: Sobald Sie sich für den Durchlauf etwas merken müssen, notieren Sie es auf einem Extrablatt (Zwischenspeicher).
- Geben Sie an, welche Zeilen vom Computer beim Durchlauf mehr als einmal gelesen und ausgeführt werden.
- Notieren Sie in einem eigenen Bereich auf Ihrem Blatt den Text, der durch die entsprechenden Methodenaufrufe ausgegeben wird.
- Vergleichen Sie Ihre Notizen für die beiden Wiederholungsarten.

Verständnishilfe: Notional Machine

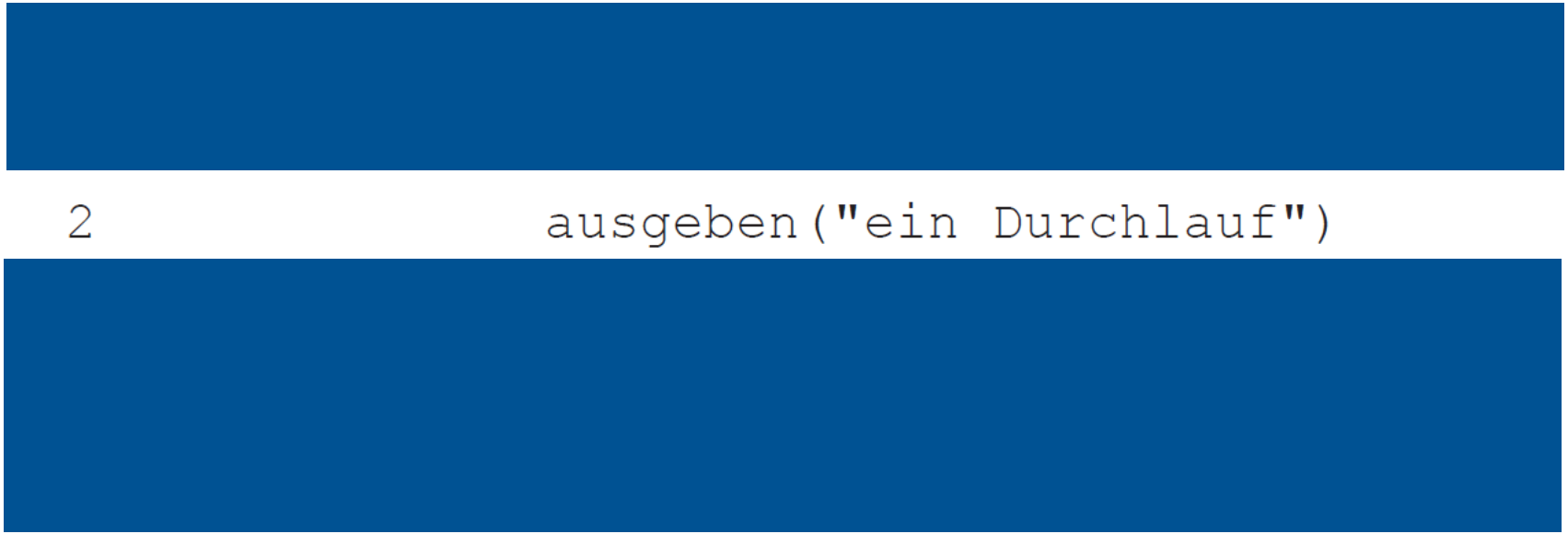
Wiederholung mit fester Anzahl

```
1       wiederhole 5 mal
2               ausgeben("ein Durchlauf")
3       endewiederhole
```

Verständnishilfe: Notional Machine

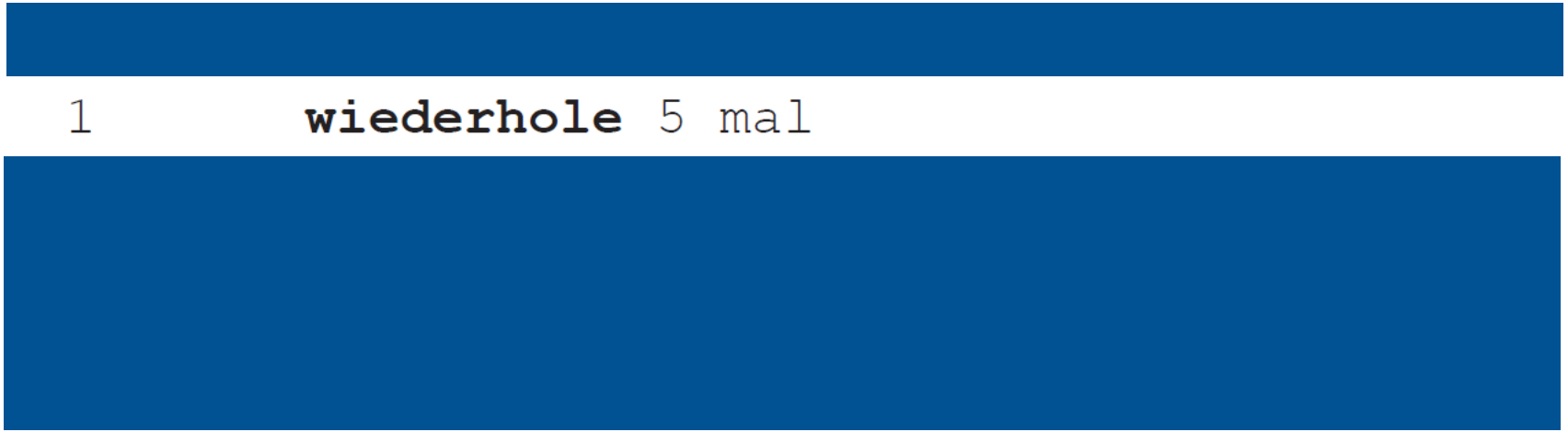


Verständnishilfe: Notional Machine

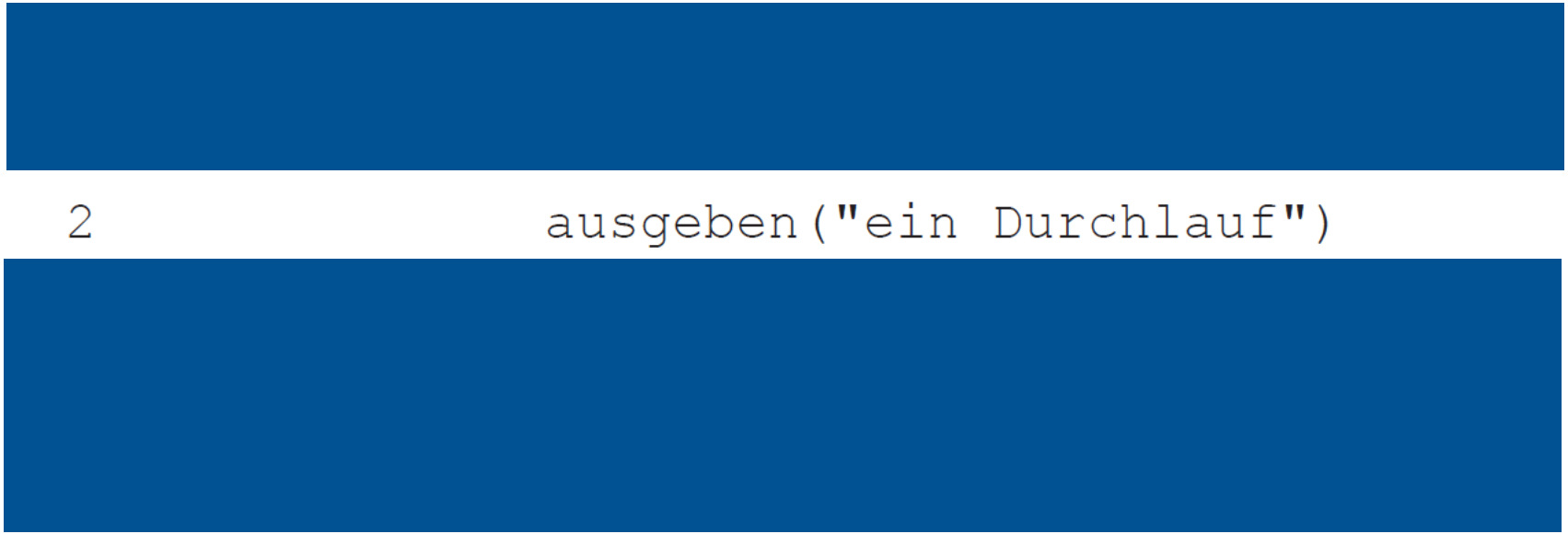


Verständnishilfe: Notional Machine

Verständnishilfe: Notional Machine



Verständnishilfe: Notional Machine



Verständnishilfe: Notional Machine

Verständnishilfe: Notional Machine

Zählwiederholung

```
1       zähle Nummer von 1 bis 6 Schritt 1
2             ausgeben(Nummer + "-ter Durchlauf")
3       endezähle
```

Verständnishilfe: Notional Machine

```
1       zähle Nummer von 1 bis 6 Schritt 1
```

Verständnishilfe: Notional Machine

```
2                ausgeben(Nummer + "-ter Durchlauf")
```

Verständnishilfe: Notional Machine

Verständnishilfe: Notional Machine

```
1       zähle Nummer von 1 bis 6 Schritt 1
```

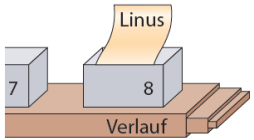
Sequenz Algorithmik 11. Jgst.

1 Algorithmen

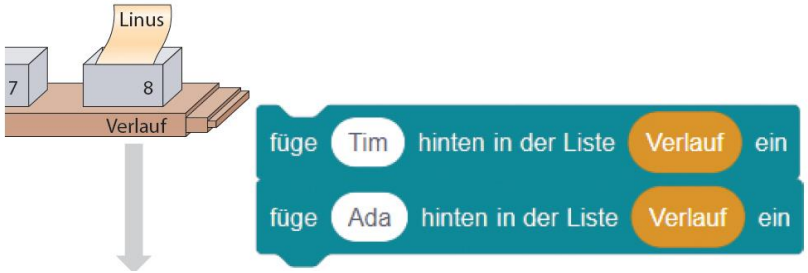
Index

Wiederholung für alle

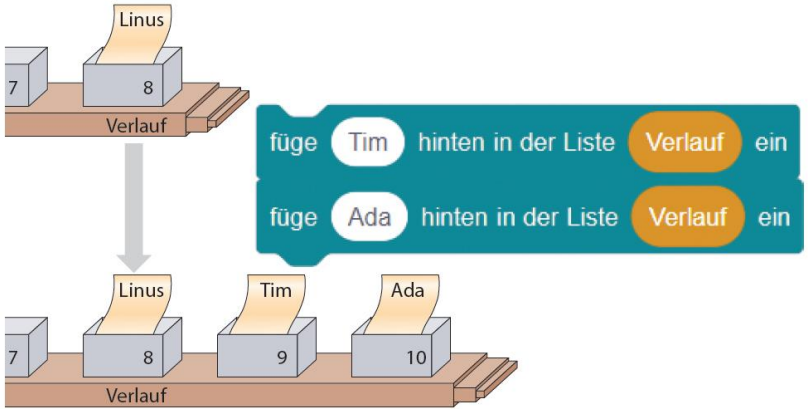
Vorstellungshilfe: Regalbrett



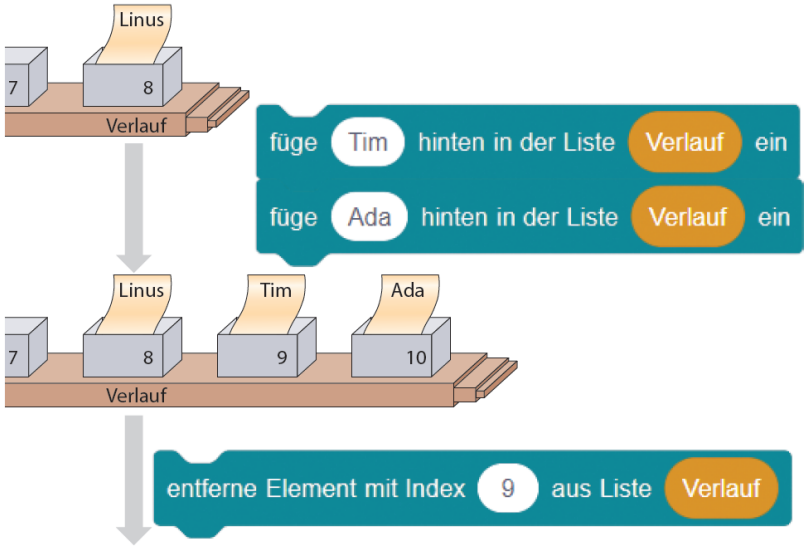
Vorstellungshilfe: Regalbrett



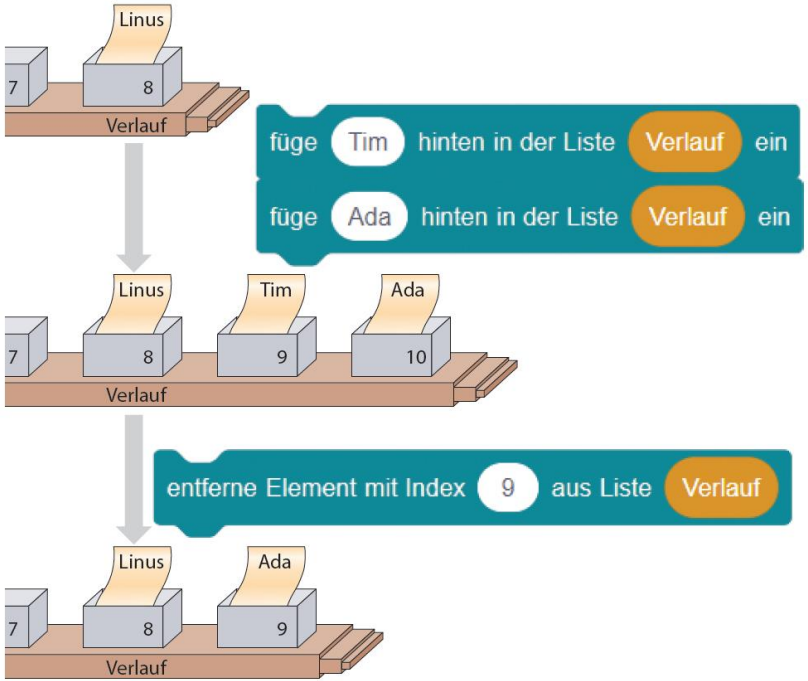
Vorstellungshilfe: Regalbrett



Vorstellungshilfe: Regalbrett



Vorstellungshilfe: Regalbrett



Sequenz Algorithmik 11. Jgst.

1 Algorithmen

Lernschwierigkeiten: Parameter, Aufruf

Methode ohne Parameter

=> Programmierer*in legt Breite und Höhe des Rechtecks mit.

Lernschwierigkeiten: Parameter, Aufruf

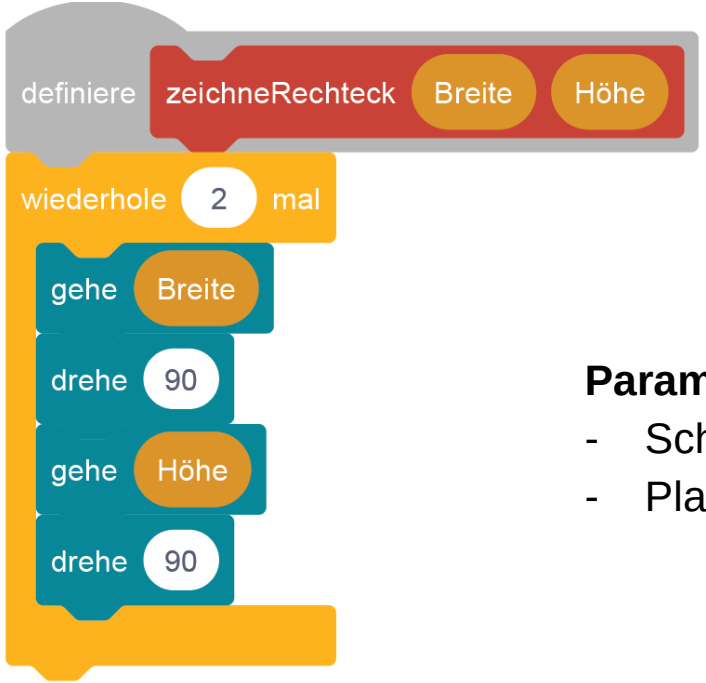
Methode ohne Parameter

=> Programmierer*in legt Breite und Höhe des Rechtecks mit.

Ziel:

Beim Methodenaufruf Informationen „mitgeben“. Aufrufer soll Breite und Höhe bestimmen können.

Lernschwierigkeiten: Parameter, Aufruf



Methode mit Parameter

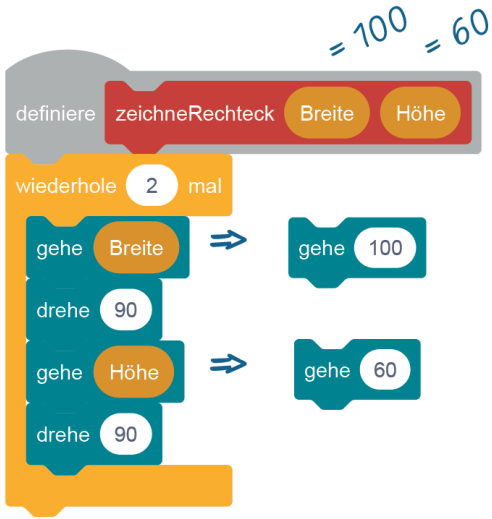
=> Aufrufer*in legt Breite und Höhe des Rechtecks mit.

Parameter

- Schnittstelle, um Werte übergeben zu können
- Platzhalter für konkrete Werte

Lernschwierigkeiten: Parameter, Aufruf

Methodenaufruf



Parameter werden durch konkrete Werte ersetzt

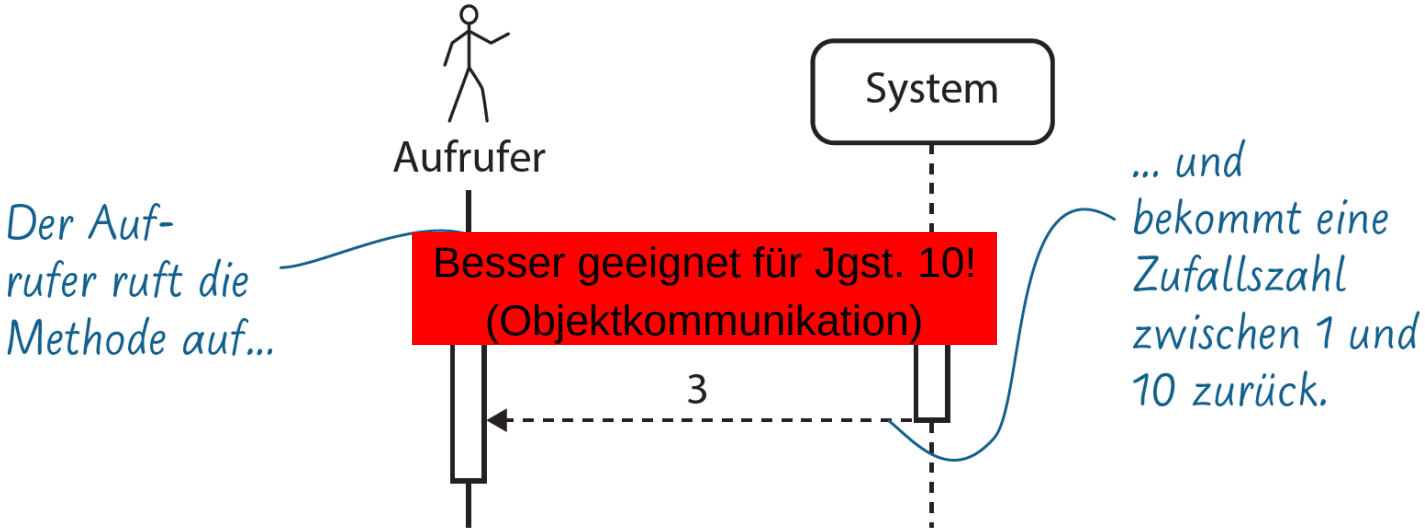
Lernschwierigkeiten: Parameter, Aufruf

```
methode zeichneRechteck()
    var Breite = 100
    var Höhe = 60
    wiederhole 2 mal
        gehe(Breite)
        drehe(90)
        gehe(Höhe)
        drehe(90)
    endewiederhole
endemethode
```

Sequenz Algorithmik 11. Jgst.

1 Algorithmen

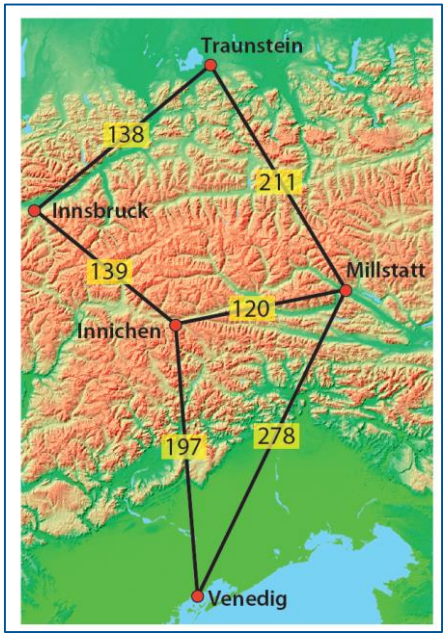
Visualisierungshilfe: Rollenspiel



Sequenz Algorithmik 11. Jgst.

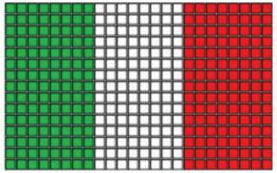
1 Algorithmen

Da steckt Informatik drin!



Pixelgrafik
Zeilenweise wird jedes Pixel einzeln über seine Farbe gespeichert.

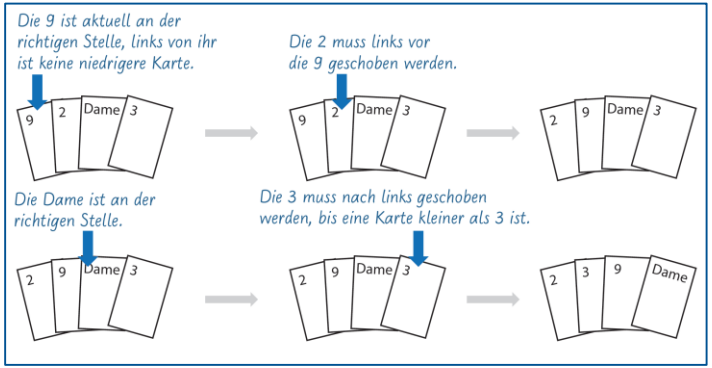
```
g g g g g g g g w w … r r
g g g g g g g g w w … r r
g g g g g g g g w w … r r
…
g g g g g g g g w w … r r
```



g steht für die Farbe „grün“, w für „weiß“ und r für „rot“.

Lauflängencodierung
Zeilenweise werden die Pixelfarbe und die Anzahl der Pixel mit der gleichen Farbe gespeichert.

```
(g, 8), (w, 8), (r,8)
(g, 8), (w, 8), (r,8)
(g, 8), (w, 8), (r,8)
 …
(g, 8), (w, 8), (r,8)
```



Java und Python Methodenbox

7 Digitale Kunst: „Flowers“ von Richard Land und Dan Cohen

Die Abbildung rechts ist inspiriert von dem Bild Flowers, das die Amerikaner Richard Land und Dan Cohen in einer Zusammenarbeit 1971 veröffentlicht haben. Das Grundelement ist ein Stern aus Linien. Über folgende Variationsmöglichkeiten können unterschiedlichste Blumenbilder erstellt werden: Strichanzahl, Strichlänge, Farbe, Winkel zwischen den Linien, Anzahl unterschiedlicher Sterne. Faszinierend an dieser Kunst ist, dass beim Einsatz von Zufallszahlen jedes Bild ein Unikat ist!

a Erstellen Sie ein Programm, das einen Stern zeichnet.

b Erweitern Sie das Programm aus a) so, dass sich bei dem Stern die einzelnen Strichlängen unterscheiden. Tipp: Eine Variable Strichlänge kann dabei hilfreich sein.

c Verändern Sie das Programm weiter, so dass ein Stern aus 30 Strichen besteht.

d Erstellen Sie Ihre Variante von „Flowers“, indem mehrere Blumen zufällig verteilt gezeichnet werden. Gerne können Sie auch die Farben variieren.

e Überlegen Sie sich eine weitere Variation und setzen Sie diese um.

AppInventor

+ motivierend
+ herzeigbares Produkt
+ kreative Rampe
+ blockbasiert
+ kein google-Account notwendig

– (Android)Smartphone wird benötigt
– Geräte müssen im selben WLAN sein
  (geht in München mit M Wlan free Wifi)
– nicht alle Konzepte äquivalent umsetzbar
  (z. B. Wiederholung), da Ereignisgesteuert

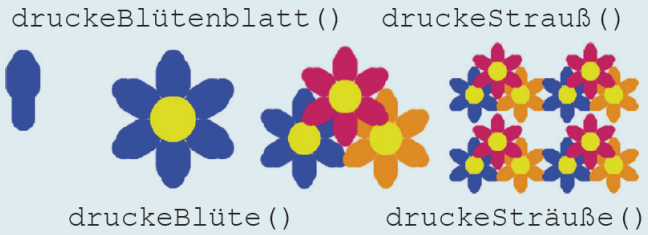
Jetzt seid ihr dran!

- Vorschlag: Pair-Programming zu zweit
- Sucht euch auf den nachfolgenden Folien Aufgaben, die euch interessieren.
- Bearbeitet die Aufgaben mit der Sprache (Snap!, AppInventor, Java, Python) eurer Wahl.

Jetzt seid ihr dran!

1.1 Erste Schritte: Arbeiten mit der Entwicklungsumgebung .....

Im gegebenen Programm sollen mit einem 3D-Drucker Anstecknadeln zur Ehrung des Schulgartenteams erstellt werden.



a Testen Sie das Programm im Team. Analysieren Sie es unter Verwendung der folgenden Begriffe, die Sie bei Bedarf unten im Text nachlesen können: Methode, Anweisung (Methodenaufruf), Sequenz.

b Variieren Sie das Programm geeignet, z. B. mehr Blütenblätter, andere Blütenform.

Jetzt seid ihr dran!

1.1 Erste Schritte: Arbeiten mit der Entwicklungsumgebung .....

6 Zeichnen mit der Turtlegrafik

Die Turtle (Schildkröte; in manchen Sprachen auch als Stift realisiert) ist eine Figur, die mit einfachen Befehlen über die Zeichenebene bewegt werden und dabei in verschiedenen Farben zeichnen kann. Zentrale Methoden einer Turtle sind:

Methode	Beschreibung
`gehe(schritte)`	bewegt die Turtle n Schritte vorwärts
`drehe(grad)`	dreht die Turtle n Grad gegen den Uhrzeigersinn (bei negativem Wert im Uhrzeigersinn)
`hebeStift()`	hebt den Stift
`senkeStift()`	senkt den Stift
`setzeStiftfarbe(farbe)`	legt eine neue Stiftfarbe fest

a Erklären Sie, was das folgende Programm leistet:

```
gehe(100)
drehe(120)
gehe(100)
drehe(120)
gehe(100)
```

Testen Sie im Anschluss Ihre Vermutung in Ihrer Programmierumgebung.

b Erstellen Sie eine neue Methode mit obigem Code im Methodenrumpf. Wählen Sie einen passenden Methodennamen.

c Stellen Sie eine Vermutung an, was die Turtle bei zweimaligem Aufruf der Methode aus b) zeichnet. Vergleichen und überprüfen Sie Ihre Vermutungen im Team.

d Entwickeln Sie selbst nach dem Vorbild der Einstiegsaufgabe aus einfacheren selbstdefinierten Methoden komplexere (z. B. Dreieck + Rechteck -> Haus -> Häuserreihe …).

Jetzt seid ihr dran!

1.1 Erste Schritte: Arbeiten mit der Entwicklungsumgebung .....

7 Gedenk-App (Projektmöglichkeit mit dem Fach Geschichte)

Wählen Sie – eventuell in Abstimmung mit Ihrer Geschichtslehrkraft – ein historisches Ereignis (in der gesamten Klasse können es auch mehrere sein), an das Sie erinnern wollen. Suchen Sie frei zugängliche Bilder und Texte, je nach Programmiersprache können auch Videos und Audioclips integriert werden.

a Entwickeln Sie eine erste einfache App, bei der Bilder und passende Texte erscheinen.

b Ergänzen Sie weitere Elemente, die nacheinander ein- und wieder ausgeblendet werden.

c Verbessern und ergänzen Sie Ihre App selbstständig in den weiteren Teilkapiteln.

Jetzt seid ihr dran!

1.1 Erste Schritte: Arbeiten mit der Entwicklungsumgebung .....

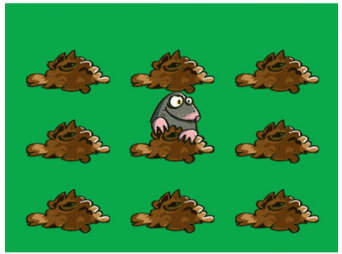
9 Fang den Maulwurf!

Ein Maulwurf hat sich neun Haufen gebaut, aus denen er zufällig auftauchen kann.

a Entwickeln Sie eine Methode, die einen Maulwurf an einer der neun Positionen zufällig für eine Sekunde erscheinen lässt. Hinweis: Deine Lehrkraft unterstützt dich bei Bedarf dabei.

b Entwickeln Sie eine weitere Methode, die die Methode aus a mehrfach aufruft.

c Entwickeln Sie eine Ereignisbehandlung für den Maulwurf, so dass das Spiel endet, sobald der Maulwurf berührt wird.



Jetzt seid ihr dran!

1.2 Speichern von Daten: Variablen .....

3 Variablen in der Entwicklungsumgebung

a Erproben Sie, wie man in der von Ihnen verwendeten Entwicklungsumgebung Variablen deklariert und initialisiert.

b Testen Sie, ob in der gleichen Variablen nacheinander Werte unterschiedlichen Datentyps gespeichert werden können. Klären Sie dazu, wie man Variablenwerte anzeigen lassen kann.

c Erstellen Sie eine Methode, die bei jeder Ausführung den Wert einer in der Variablen gespeicherten Zahl um 1 erhöht bzw. verdoppelt.

d Erkunden Sie, welche Operationen und Methoden die Programmiersprache für die unterschiedlichen Datentypen bereitstellt, und testen Sie diese.

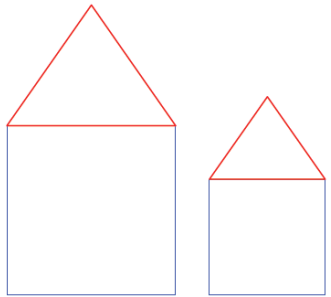
Jetzt seid ihr dran!

6 Zeichnen mit der Turtlegrafik (Teil 2)

Die Turtle soll nun flexibler werden. In einer Variablen soll ein Wert für den aktuell gültigen Maßstab (z. B. 70 Pixel) gespeichert werden. Die Turtle soll dann verschiedene Figuren flexibel abhängig vom Maßstab zeichnen können.

a Entwickeln Sie zwei Methoden *QuadratZeichnen()* und *DreieckZeichnen()*, die ein Quadrat bzw. ein gleichseitiges Dreieck jeweils in der passenden Größe zeichnen.

b Verbinden Sie die beiden Methoden in einer Methode *HausZeichnen()*, so dass die Turtle ein Haus variabler Größe zeichnen kann. Testen Sie für verschiedene Werte der Variablen.

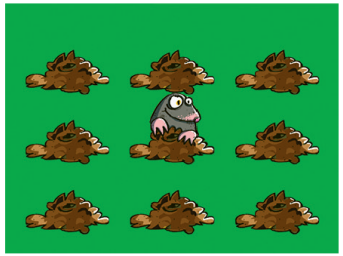


Jetzt seid ihr dran!

1.2 Speichern von Daten: Variablen .....

8 Fang den Maulwurf! (Teil 2)

Das Maulwurfspiel soll verbessert werden. Der Spieler oder die Spielerin hat zu Beginn 10 Punkte. Dieser Punktestand soll in einer Variablen gespeichert werden. Jedes Mal, wenn der Maulwurf auftaucht, wird ein Punkt abgezogen. Bei 0 Punkten endet das Spiel.



Jetzt seid ihr dran!

1.5 Gleiches öfter ausführen: Wiederholungen .....

Aufgabe nur für Snap!

Tinas Tanzschule erstellt für die Grundschritte kurze Filme, welche aus sich fortlaufend wiederholenden Bildfolgen bestehen. Dazu werden Standbilder eines tanzenden Paars animiert. Mit diesen Animationen können die einzelnen Tanzschritte detailliert nachvollzogen werden.

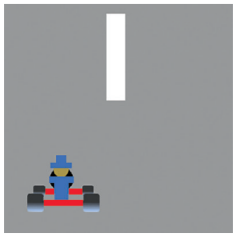
a Vergleichen Sie die beiden Lösungen der Animation. Ändern Sie die Schrittweite der Tanzenden in beiden Lösungen und geben Sie anschließend stichwortartig zwei Vorteile der kürzeren Lösung an.

b Die Schülerinnen und Schüler wollen gerne die einzelnen Tanzschritte nachvollziehen. Daher erstellt Tina eine Animation, welche zu jedem Taktschlag die aktuelle Tanzpose enthält und die Zählzeit angibt. Die beiden Lösungen in der Vorlage enthalten je zwei unterschiedliche Wiederholungen. Vergleichen Sie beide. Begründen Sie, welche der beiden hinsichtlich einer beliebigen Startposition besser geeignet ist.

c Für Schnelle: Eine dritte Wiederholungsart ist die Zählwiederholung. Vergleichen Sie die Varianten aus b) mit der Variante in dieser Vorlage. Erklären Sie, warum i Zählvariable genannt wird.

Jetzt seid ihr dran!

1.5 Gleiches öfter ausführen: Wiederholungen .....



4 Autorennen mit bewegtem Hintergrund

Eine Technik beim Entwickeln von Spielen ist es, den Hintergrund so zu programmieren, dass dieser sich bewegt. Auf diese Weise sieht es so aus, als würde sich eine Figur bewegen. In der Vorlage bewegt sich ein Mittelstreifen über eine Fahrbahn.

a Analysieren Sie das Programm. Verändern Sie die Werte in den bedingten Wiederholungen und testen Sie jeweils die Auswirkung.

b Modifizieren Sie das Programm so, dass sich das Fahrzeug scheinbar schneller bewegt, sich also der Mittelstreifen schneller bewegt.

c Führen Sie eine Variable Level ein, welche am Anfang auf 0 gesetzt wird. Formulieren Sie die Änderung der y-Position in der Bewegung der Mittelstreifen in Abhängigkeit vom Wert von Level. Wenn Level den Wert 0 hat, soll die y-Position um –3 geändert werden. Beim Wert 1 um –4, bei 2 um –5, …

d Analysieren Sie die Wirkung des nebenstehenden Blocks.

Lassen Sie am Anfang des Programms die Stoppuhr starten und fügen Sie den Block an der Stelle ein, an der der Mittelstreifen von oben startet.

Fortsetzung
nächste Folie

Jetzt seid ihr dran!

1.5 Gleiches öfter ausführen: Wiederholungen .....

e Implementieren Sie die Bewegung für zwei weitere Figuren (z. B. Blume und Regentropfen) so, dass diese sich wie der Mittelstreifen von oben nach unten bewegen. Erreicht eine der Figuren den unteren Rand, so soll sie erst nach einer gewissen Zeit wieder am oberen Rand auftauchen. Verwenden Sie dazu je nach Entwicklungsumgebung entweder eine zufällige Wartezeit oder lassen sie die Figur bis zu einer zufälligen y-Position außerhalb des Spielfelds gleiten.

f Für Schnelle: Erweitern Sie das Programm so, dass das Fahrzeug eine bestimmte Figur (z. B. den Regentropfen) nicht berühren darf. Fügen Sie ggf. eine Variable Leben ein, die mit einem Startwert (z. B. 3) belegt wird und bei jeder Berührung um 1 verringert wird. Wenn Leben den Wert 0 erreicht hat, soll das Programm stoppen.

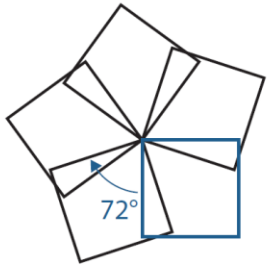
g Für ganz Schnelle: Erweitern Sie das Programm so, dass bei Berührung mit der Blume der Wert der Variablen Leben um eins erhöht wird.

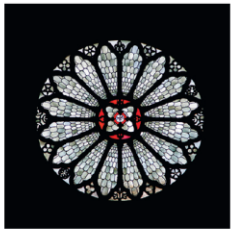
Jetzt seid ihr dran!

1.5 Gleiches öfter ausführen: Wiederholungen .....

6 Digitale Kunst: Rosetten

Blattrosetten sind kreisförmig dicht angeordnete Blätter. In unterschiedlichen Epochen haben Kunstschaffende Rosetten in ihren Werken aufgegriffen. Mit einer Turtle (oder einer anderen Figur mit Zeichenfunktion) können Rosetten gezeichnet werden, indem wiederholt eine einfache geometrische Figur gezeichnet und dann die Drehrichtung verändert wird. Die Abbildung rechts zeigt ein Beispiel, bei dem die Basisfigur, ein Quadrat, fünfmal gezeichnet wird.





a Begründen Sie, warum der Drehwinkel 72° gewählt wurde.

b Analysieren Sie das Programm in der Vorlage. Erhöhen Sie die Lesbarkeit des Programms, indem Sie einen Teil der Anweisungen in einer Methode zusammenfassen. Zeichnen Sie dann andere Rosetten durch das Verändern des Drehwinkels und der Anzahl der Wiederholungen. Speichern Sie eine der Rosetten als eigenen Methodenblock mit dem Namen *zeichneRosette1*.

c Verändern Sie nun die Basisform auf z. B. ein Achteck. Speichern Sie wieder als Methodenblock.

d Überlegen Sie eine eigene Variation (z. B. mit Farbe) und setzen Sie diese um.

Jetzt seid ihr dran!

1.8 Eine Antwort erhalten: Methoden mit Rückgabewert .....

Ein einfacher Spielautomat hat drei Walzen mit sieben unterschiedlichen Symbolen. Die Steuerung wird mit einer Software betrieben.

a Erkunden Sie das Programm und geben Sie an, wie viel ein Spiel kostet und wie groß der Gewinn bei drei gleichen Symbolen ist.

b Bedingungen kann man als Methoden interpretieren, welche dem Aufrufer Informationen zurückgeben. Beschreiben Sie in eigenen Worten, welche Information der Aufrufer bei *istGroßerGewinn* bekommt. Geben Sie auch den Datentyp des Rückgabewerts an.

c Bei nur zwei gleichen Symbolen soll ein kleiner Gewinn ausgeschüttet werden, der das Guthaben um zwei erhöht. Implementieren Sie zunächst eine Methode *istKleinerGewinn*. Verändern Sie die Methode *ermittleGewinn* mithilfe dieser Methode, so dass auch bei einem kleinen Gewinn das Guthaben verändert wird.

d Spielen Sie ein paar Runden. Begründen Sie anhand des Spielautomaten, welcher Reiz vom Glückspiel ausgeht, der in eine Sucht führen kann.

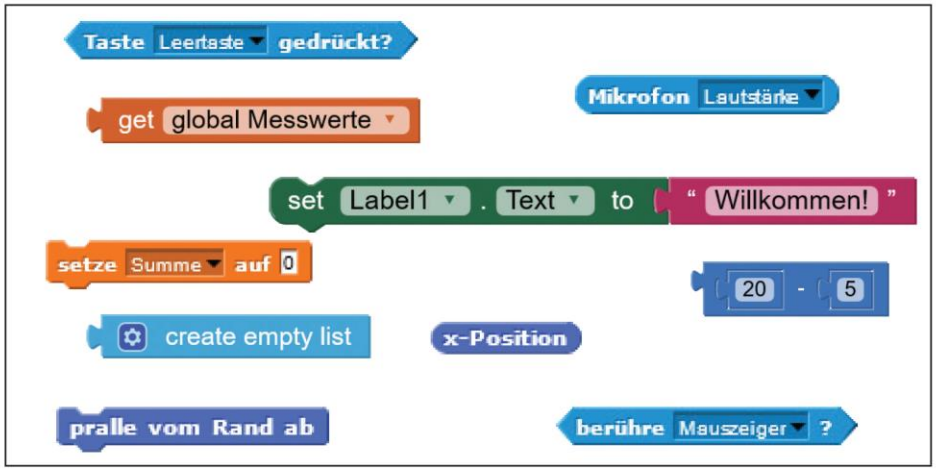
e Für Schnelle: Eines der Symbole ist ein rotes Kreuz. Verändern Sie das Programm so, dass bei zwei roten Kreuzen ein kleiner Betrag, bei drei roten Kreuzen ein großer Betrag vom Guthaben abgezogen wird.

Jetzt seid ihr dran!

1.8 Eine Antwort erhalten: Methoden mit Rückgabewert .....

3 Rückgabe oder keine Rückgabe, das ist hier die Frage

a Betrachten Sie folgende Quellcodeausschnitte. Geben Sie jeweils an, ob es sich um eine Methode mit Rückgabe handelt oder nicht. Vergleichen Sie Ihre Meinung mit der Ihrer Nachbarin oder Ihres Nachbarn. Testen Sie wo nötig und falls möglich in Ihrer Programmiersprache.



b Erstellen Sie für eine blockbasierte Programmiersprache (z. B. Snap!, ...) eine Übersicht unterschiedlicher Blockformen und deren Bedeutung, z. B. ⬡ sind Bedingungen bzw. Methoden, … .

Jetzt seid ihr dran!

1.8 Eine Antwort erhalten: Methoden mit Rückgabewert .....

5 Die passende Anrede

In vielen Portalen, z. B. beim Onlinebanking oder -shopping wird man nach der Anmeldung individuell begrüßt, z. B.: „Guten Tag, Frau Mayer."

a Entwickeln Sie eine Methode mit Rückgabe, die als Eingabeparameter den Namen und das Geschlecht erwartet und daraus eine passende Zeichenkette generiert und zurückgibt.

b Nutzen Sie die Methode in einer grafischen Benutzeroberfläche zur Erstanmeldung mit der Möglichkeit zur Auswahl von Geschlecht und Eingabe des Namens, welche die Methode aus a) anschließend zur Ausgabe des Wertes nutzt.

c Verbessern Sie die Methode aus a) um weitere Features (z. B. spezielle Begrüßung morgens und abends oder bei Kunden aus einer bestimmten Region („Grüß Gott", Postleitzahlbereich!) oder altersabhängig („Hi, Tom!").