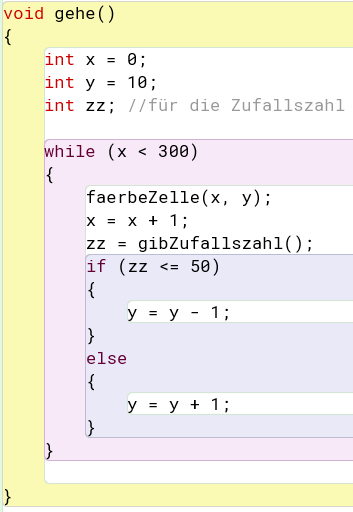
6.5 Algorithmen modellieren und implementieren

Aufgabe 1

1. siehe Projekt



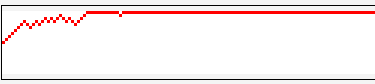
6-5 Random Walk - Loesung Aufgabe 1

1. siehe Projekt

6-5 Random Walk - Loesung Aufgabe 1

Ändert man die Wahrscheinlichkeit für die Änderung der y-Koordinate nach oben, so verlaufen deutlich mehr Zufallswege so, dass sie an der oberen Kante angrenzen. Die Änderung erfolgt durch eine Anpassung der Zahl 50 in z. B. 60 in Zeile 32 des Programmcodes.

Beispiel:



Aufgabe 2

1. siehe Projekt 6-5 Random Walk – Loesung Aufgabe 2

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungImplementierung:

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungDie Variable y muss auf den Wert gesetzt werden, der der halben Höhe entspricht:

int y = hoehe/2;

Die bedingte Wiederholung darf nur durchgeführt werden, solange x noch nicht so groß ist wie die Breite der Welt:

while (x < breite)

1. Liefert die Zufallszahl häufig Werte von über 50 bzw. unter 50, so wird der untere bzw. obere Rand der Welt überschritten. Greenfoot setzt den Punkt dann an den Rand der Welt.

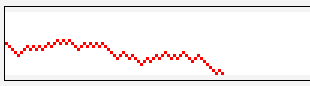
Um beim Erreichen des Randes (y < 0 || y > hoehe + 1) die Wiederholung abzubrechen, kann z. B. in diesem Fall x auf einen Wert gesetzt werden, der größer ist als die Breite der Welt, z. B. x = breite + 1

Dann wird die bedingte Wiederholung (siehe 2a) verlassen.

siehe Projekt 6.5 Random Walk – Loesung Aufgabe 2b

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungImplementierung:



1. siehe Projekt

6-5 Random Walk – Loesung Aufgabe 2c1

Ein Bild, das Tisch enthält.

Automatisch generierte BeschreibungImplementierung:

Liegt die Zufallszahl zwischen 30 (ausschließlich) und 70, so ändert sich sich die y-Koordinate nicht.

siehe Projekt 6-5 Random Walk – Loesung Aufgabe 2c2

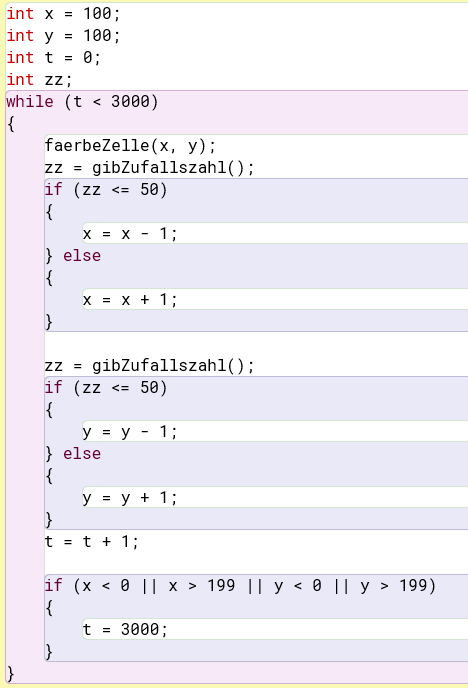
Implementierung:

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Aufgabe 3

1. siehe Projekt 6-5 Random Walk – Loesung Aufgabe 3

Implementierung:

1. Um den Programmablauf zu stoppen, könnte man eine weitere Variable t einführen, die bei jedem Zeichnen eines Kästchens um 1 erhöht wird. Dann kann man mit der bedingten Wiederholung while (t < AnzahlKästchen) festlegen, nach wie vielen Schritte das Programm stoppt.

siehe Projekt 6-5 Random Walk – Loesung Aufgabe 3

Implementierung: siehe Aufgabe 3a)