

Kontrollstrukturen

Verzweigungen und Schleifen

Inhaltsverzeichnis

Rock Paper Scissors.....	1
Bisektionsverfahren.....	2
Betrunkener.....	2
Halbieren.....	3

Rock Paper Scissors

Schreiben Sie ein Java-Programm „RockPaperScissors“, das ein Spiel Schere-Stein-Papier gegen den Computer durchführt. Der Computer soll sich ein Symbol „ausdenken“ und anschließend soll der Benutzer die Möglichkeit haben eines der drei Symbole zu wählen. Die Ausgabe soll angeben, wer gewonnen hat und welche Symbole verwendet wurden.

Tipp: Falls Sie die Regeln des Spiels nicht kennen, können Sie diese leicht online recherchieren. Mit dem Befehl `(int) (Math.random() * 3 + 1)` erzeugen Sie eine ganzzahlige Zufallszahl im Bereich von 1 bis 3.

Auf der Konsole könnte das beispielsweise so aussehen:

```

1 = Rock (Stein)
2 = Paper (Papier)
3 = Scissors (Schere)
Welches Symbol wollen Sie nutzen? 2

Der Computer nutzt Rock.
Sie nutzen Paper.
Sie gewinnen.
  
```

Erweitern Sie Ihr Programm, sodass nicht nur eine Runde gespielt wird. Verwenden Sie dafür eine Schleife. Zählen Sie den aktuellen Punktestand mit und geben Sie ihn aus.

Bisektionsverfahren

Mit Hilfe des Bisektionsverfahrens lassen sich Nullstellen von stetigen Funktionen näherungsweise bestimmen. Es startet mit einem Intervall, innerhalb dessen die Funktion genau einen Vorzeichenwechsel vollzieht. Da die Funktion innerhalb des Intervalls einen Vorzeichenwechsel vollzieht, muss sich die Nullstelle innerhalb des Intervalls befinden.

Beispiel:

$$f(x) = x^3 + 3x^2 - 1$$

$$f(0) < 0, \quad f(1) > 0$$

Intervall:

$$[0, 1]$$

Mitte des Intervalls: 0.5

$$f(0.5) = -0.125 < 0$$

Intervall (untere Grenze ersetzen):

$$[0.5, 1]$$

Mitte des Intervalls: 0.75

$$f(0.75) = 1.109375 > 0$$

Intervall (obere Grenze ersetzen):

$$[0.5, 0.75]$$

Mitte des Intervalls: 0.625

$$f(0.625) = 0.416015625 > 0$$

Intervall (obere Grenze ersetzen):

$$[0.5, 0.625]$$

...

Das Bisektionsverfahren halbiert das Intervall in jedem Schritt und wählt dasjenige der beiden Teilintervalle aus, in dem der Vorzeichenwechsel stattfindet, also in dem die Nullstelle liegt. Das Intervall wird so lange halbiert bis es eine vorgegebene Größe unterschreitet. Die Nullstelle ist bis auf die Ungenauigkeit der Intervallgröße gefunden.

Schreiben Sie ein Java-Programm, das mit Hilfe des Bisektionsverfahrens die Nullstelle der Funktion $f(x) = x^3 + 3x^2 - 1$ im Intervall $[0,1]$ näherungsweise bestimmt. Die Funktion hat außerdem zwei weitere Nullstellen im negativen Bereich. Schreiben Sie das Programm aber nur für die Nullstelle im Intervall $[0,1]$.

Betrunkener

Max Meier studiert Informatik. Er wohnt im Studentenwohnheim in Zimmer 1. Er geht in seinem Wohnheim abends gerne in die hauseigene Bar (Zimmer 7). Nachdem er jedoch einige Biere getrunken hat, erinnert er sich nicht mehr an seine Zimmernummer. Doch

Max hat sich einen Algorithmus ausgedacht, der ihn garantiert irgendwann zu seinem Zimmer führt:

Er wirft eine Münze. Bei Kopf geht er eine Zimmernummer nach oben, bei Zahl eine Zimmernummer nach unten. An diesem Zimmer probiert er seinen Schlüssel. Wenn der Schlüssel passt, ist er in seinem Zimmer angekommen. Passt er nicht, geht es wieder mit dem Münzwurf weiter.

Zum Beispiel:

```
Zimmer 7 → Kopf → Zimmer 8 → Zahl → Zimmer 7 → Kopf →  
Zimmer 8 → Zahl → Zimmer 7 → Zahl → Zimmer 6 → Zahl →  
Zimmer 5 → Kopf → Zimmer 6 → usw.
```

Schreiben Sie ein Java-Programm, das die Route von Max Meier simuliert. Wie oft muss er seinen Schlüssel in ein Schloss stecken bis er an seinem Zimmer ist?

Sie können das Problem auch variieren, indem Sie Max in Zimmer 17 oder sogar in Zimmer 117 starten lassen. Kommt er mit seiner Taktik wirklich immer wieder zurück?

Halbieren

Schreiben Sie ein Java-Programm „HalveDouble“, das eine double-Variable `x` deklariert und mit 1 initialisiert. Nun soll `x` so lange halbiert werden bis der Wert auf exakt 0 gesunken ist, also bis die Prüfung `x == 0` auch `true` ergibt. Das Programm soll dann ausgeben, wie oft halbiert wurde. Schreiben Sie ein weiteres Java-Programm

„HalveFloat“, das eine float-Variable `x` deklariert und mit 1 initialisiert. Auch das Programm soll `x` so lange halbieren bis der Wert 0 ist und die Anzahl der Halbierungen ausgeben.

Erklären Sie, warum der Wert nur durch Halbierungen jeweils überhaupt auf 0 sinken kann. Und erklären Sie, wie die Anzahl an Halbierungen zustande kommt.