

Programmbeschreibung und Flussdiagramm

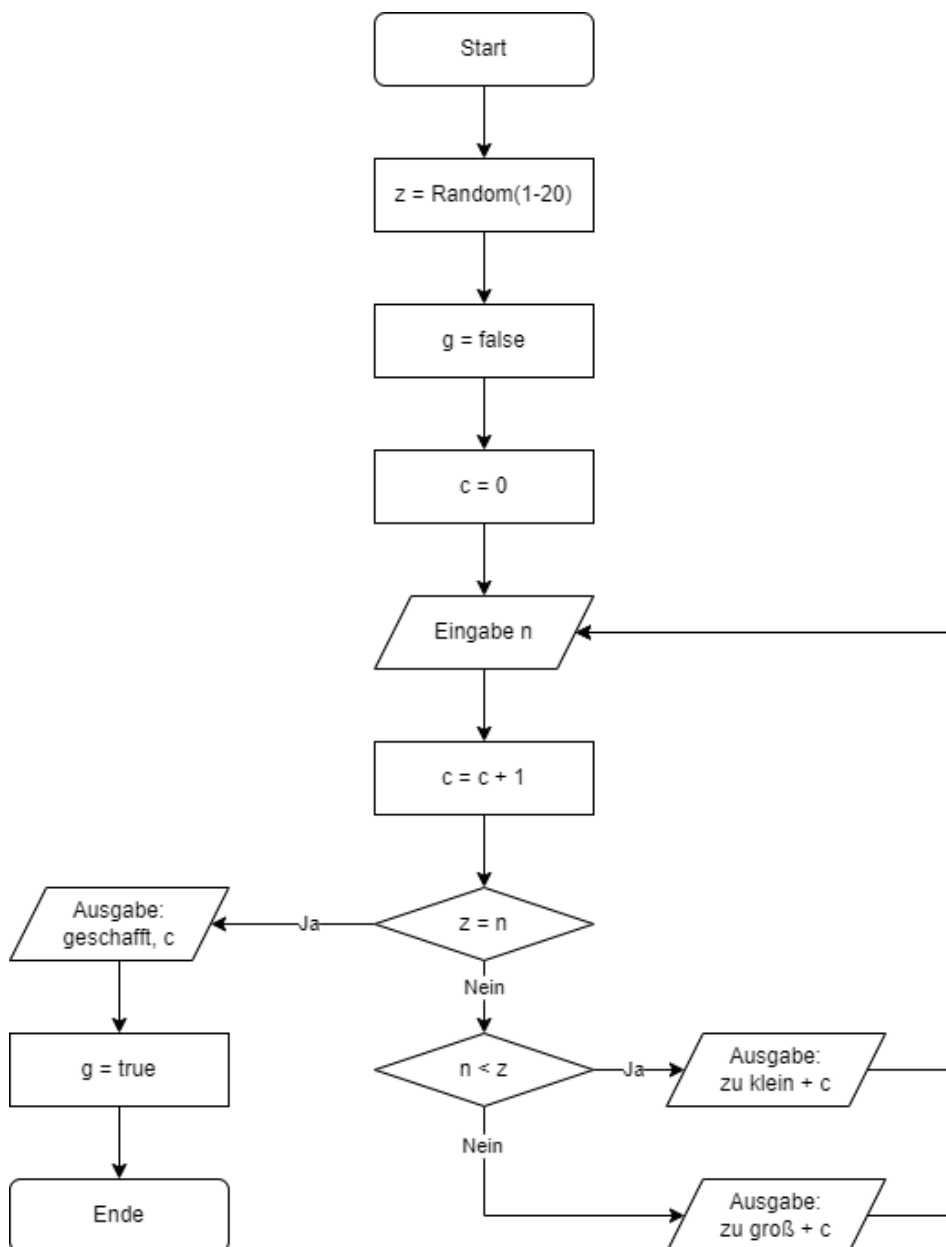


Hinweis

Dieses Programm „denkt“ sich eine zufällige Zahl zwischen 1 und 20 aus. Der Nutzer soll diese Zahl erraten. Das Programm gibt die Anzahl der Versuche wieder.

Die Umsetzung des Programm verwendet:

- while-Schleife
- if-Abfrage
- Variablen
- Boolean (wahr/falsch)
- Inkrementieren



Schritt-für-Schritt-Anleitung

In dem folgenden Teil führt „Schritt-für-Schritt“ durch das Flussdiagramm.

Am Ende dieser Anleitung steht das vollständige Programm passend zum Flussdiagramm.

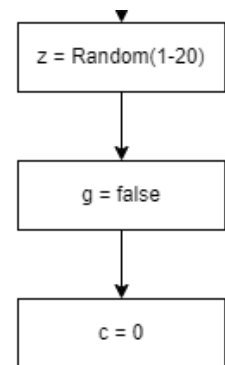
1. Schritt: Die Eingabe (Zeile 1 und 3)

„let“ deklariert die Variablen z, g und c.

Die Variable c dient zum „Hochzählen“, um die Länge der Reihe einzuhalten.

JavaScript

```
1 let z = Math.floor(Math.random() * 20) + 1;  
2 let g = false;  
3 let c = 0;
```



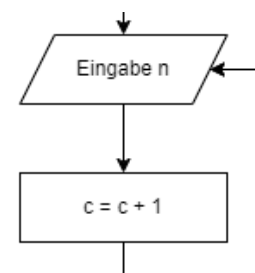
2. Schritt: Die while-Schleife (Zeile 5 bis 7)

Solange **g nicht wahr** ist, die Frage nach der richtigen Zahl gestellt.

Die Variable **a** wird mit dem Wert aus "**prompt**" deklariert. Der Zähler c erhöht sich um eine Einheit.

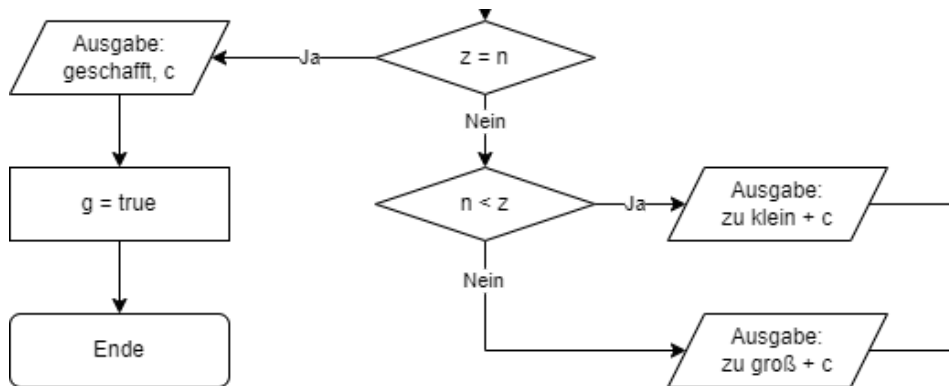
JavaScript

```
1 let z = Math.floor(Math.random() * 20) + 1;  
2 let g = false;  
3 let c = 0;  
4  
5 while (!g) {  
6   let n = prompt('Gebe die Zahl an: ');  
7   c++;
```



3. Schritt: Die Abzweigungen (Zeile 8 bis 16)

- Ist $z = n$, dann wird g auf wahr gestellt. Zuvor gibt es die Ausgabe „geschafft“ mit dem Zählerwert.
- Ist $n < z$, dann wird die Ausgabe „zu klein“ mit dem Zähler ausgegeben.
- Ist $n > z$, dann wird die Ausgabe „zu groß“ mit dem Zähler ausgegeben.



JavaScript

```
1 let z = Math.floor(Math.random() * 20) + 1;
2 let g = false;
3 let c = 0;
4
5 while (!g) {
6   let n = prompt('Gebe die Zahl an: ');
7   c++;
8   if (z == n) {
9     alert('geschafft ' + c);
10    g = true;
11  } else if (n < z) {
12    alert('zu klein ' + c);
13  } else if (n > z) {
14    alert('zu groß ' + c);
15  }
16 }
```

ENDE

Aufgaben

- ① Füge in dem Quellcode passend zu den Schritten Kommentare ein, die den jeweiligen Teil in seiner Funktion beschreiben.

JavaScript

```
1 // Zwei "Balken" ergeben einen Zeilenkommentar
2
3 /* Balken und Sternchen am Anfang und am Ende
4 ergeben einen Kommentar auf mehreren Zeilen */
```

- ② Überlege Dir mögliche Verbesserungen für das Programm. Beispielsweise:

- Könnte der Text der Eingabe und der Ausgabe verbessert werden.
- Lässt sich der Schwierigkeitsgrad erhöhen?
- Könnte die letzte Else-If-Abfrage durch Else ersetzt werden?

- ③ Passe das Flussdiagramm und den Quelltext entsprechend Deiner Überlegungen aus Aufgabe 2 an.