

## if, switch, while, do-while, for

### Aufgabe 1: NumberSearch

Schreiben Sie ein Programm, das eine vom Benutzer ausgedachte ganze Zahl von 0 bis 100 möglichst schnell errät. Das Programm gibt eine Vermutung aus und der Benutzer antwortet mit:

- '<' wenn die ausgedachte Zahl kleiner ist,
- '>' wenn die ausgedachte Zahl größer ist,
- '=' wenn die Vermutung stimmt.

Geben Sie die Anzahl der Vermutungen aus, die notwendig waren, um die ausgedachte Zahl des Benutzers zu erraten.

### Hinweise

Überlegen Sie sich eine gute Strategie für die Vermutungen (welche ist eine gute?) und treffen Sie Vorkehrungen, damit auch alle Zahlen, insbesondere 0 und 100, gefunden werden. Verwenden Sie für das Einlesen der Zeichen '<', '>' und '=' die Methode *In.readChar()*. Diese überliest Leerzeichen und gibt das erste Zeichen nach den Leerzeichen zurück.

Gehen Sie bei der Lösung wie folgt vor:

1. Beschreiben Sie die Lösungsidee zu dem Algorithmus.
2. Stellen Sie den Algorithmus in Prosa dar.
3. Realisieren Sie den Algorithmus in Java.
4. Führen Sie zur Überprüfung der Funktion sinnvolle Ausgaben von aktuellen Werten an interessanten Stellen des Programms durch, z.B. am Anfang und Ende der Schleife.
5. Stellen Sie einen Testplan auf, d.h. definieren Sie interessante Testfälle und geben Sie die erwarteten Ergebnisse an.
6. Testen Sie Ihr Programm nach diesem Testplan.

## Lösungen zu den Aufgaben

### Lösung 1: NumberSearch

#### Lösungsidee

Man verwendet eine untere und obere Schranke, die den Bereich definieren, in dem sich die gesuchte Zahl befinden muss. Dann gibt man als Vermutung jene Zahl aus, die in der Mitte dieses Bereichs liegt. Vom Benutzer bekommt man den Hinweis, ob die Zahl kleiner oder größer als die Vermutung ist. Ist die Vermutung zu klein, verschiebt man die untere Grenze bis zur Vermutung; ist sie zu groß, verschiebt man die obere Grenze. Dies wiederholt man so lange, bis die Zahl gefunden wurde, oder die untere und obere Schranke gleich ist, womit man die gesuchte Zahl automatisch gefunden hat.

#### Prosabeschreibung

```
SETZE untere Schranke auf 0
SETZE obere Schranke auf 100
SETZE Anzahl Versuche auf 0
SETZE gefunden auf falsch
GIB "Denke Zahl aus" AUS

WIEDERHOLE
  ERHÖHE Anzahl Versuche um 1
  Wenn untere Schranke = obere Schranke
    SETZE gefunden auf wahr
    SETZE Vermutung auf eine der Schranken
  Sonst
    BERECHNE Vermutung = (untere Schranke + obere Schranke) / 2
    GIB Vermutung AUS

WIEDERHOLE
  LIES Antwort (<, >, =) EIN
  SOLANGE Antwort nicht gueltig

  Wenn Antwort <
    SETZE obere Schranke auf Vermutung - 1
  Sonst Wenn Antwort >
    SETZE untere Schranke auf Vermutung + 1
  Sonst Wenn Antwort =
    SETZE gefunden auf wahr
  SOLANGE gefunden falsch ist
  GIB Vermutung und Anzahl der Versuche AUS
```

## Source Code

```
1  class NumberSearch {
2      static final boolean DEBUG_OUTPUT = false;
3
4      public static void main(String[] args) {
5          int lowerBound = 0;
6          int upperBound = 100;
7          int cntGuesses = 0;
8
9          Out.println(
10             String.format(
11                 "Bitte denken Sie sich eine Zahl zwischen %d und %d aus!",
12                 lowerBound, upperBound));
13         boolean hasFound = false;
14         int guess;
15
16         do {
17             cntGuesses++;
18
19             if (lowerBound == upperBound) {
20                 hasFound = true;
21                 guess = lowerBound;
22             } else {
23                 guess = (lowerBound + upperBound) / 2;
24                 Out.println(
25                     String.format(
26                         "Meine %d. Vermutung ist: %d",
27                         cntGuesses, guess));
28
29                 char reply;
30                 boolean isValidReply;
31
32                 do {
33                     Out.print("Bitte gib eine Antwort ein (<, >, =): ");
34                     reply = In.readChar();
35                     isValidReply = (reply == '<' || reply == '>' || reply == '=');
36
37                     if (!isValidReply) {
38                         Out.println("Ungültige Eingabe");
39                     }
40                 } while (!isValidReply);
41
42                 if (DEBUG_OUTPUT) {
43                     Out.println(reply);
44                 }
45
46                 switch (reply) {
47                     case '<':
48                         upperBound = guess - 1;
49                         break;
50                     case '>':
51                         lowerBound = guess + 1;
52                         break;
53                     case '=':
54                         hasFound = true;
55                         break;
56                 }
57
58                 if (DEBUG_OUTPUT) {
59                     Out.println(
60                         String.format(
```

```
61         "lowerBound=%d, upperBound=%d",
62         lowerBound, upperBound));
63     }
64 }
65 } while (!hasFound);
66
67 Out.println(
68     String.format(
69         "Die Zahl %d im %d. Versuch gefunden.",
70         guess, cntGuesses));
71 }
72 }
```

## Testausgaben

Einige interessante Daten werden vom Programm bereits ausgegeben, z.B. die aktuelle Anzahl der Vermutungen, oder die nächste Vermutung. Zusätzlich interessant sind jedoch die aktuellen Schranken. Diese werden über einen optionalen Ausdruck ausgegeben. Über die Konstante `DEBUG_OUTPUT` lässt sich die Ausgabe ein- bzw. ausschalten.

**Tabelle 1:** Testfälle

#	Beschreibung	Gesucht	Erwartetes Ergebnis	Status
1	untere Schranke des Suchraums	0	6 Vermutungen, untere Schranke = 0, obere Schranke = 1	OK
2	obere Schranke des Suchraums	100	7 Vermutungen, untere Schranke = 100, obere Schranke = 100	OK
3	Mitte des Suchraums	50	1 Vermutung, untere Schranke = 0, obere Schranke = 100	OK
4	Regelfall	22	6 Vermutungen, untere Schranke = 21, obere Schranke = 23	OK
5	Spezialfall (obere == untere Schranke)	21	7 Vermutungen, untere Schranke = 21, obere Schranke = 21, Zahl bei 7. Vermutung klar	OK

### Testfall #1 (gesucht ist 0)

```
Bitte denken Sie sich eine Zahl zwischen 0 und 100 aus!
Meine 1. Vermutung ist: 50
Bitte gib eine Antwort ein (<, >, =): <
lowerBound=0, upperBound=49
Meine 2. Vermutung ist: 24
Bitte gib eine Antwort ein (<, >, =): <
lowerBound=0, upperBound=23
Meine 3. Vermutung ist: 11
Bitte gib eine Antwort ein (<, >, =): <
lowerBound=0, upperBound=10
Meine 4. Vermutung ist: 5
Bitte gib eine Antwort ein (<, >, =): <
lowerBound=0, upperBound=4
Meine 5. Vermutung ist: 2
Bitte gib eine Antwort ein (<, >, =): <
lowerBound=0, upperBound=1
Meine 6. Vermutung ist: 0
Bitte gib eine Antwort ein (<, >, =): =
lowerBound=0, upperBound=1
Die Zahl 0 im 6. Versuch gefunden.
```

### Testfall #2 (gesucht ist 100)

Bitte denken Sie sich eine Zahl zwischen 0 und 100 aus!  
Meine 1. Vermutung ist: 50  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=51, upperBound=100  
Meine 2. Vermutung ist: 75  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=76, upperBound=100  
Meine 3. Vermutung ist: 88  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=89, upperBound=100  
Meine 4. Vermutung ist: 94  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=95, upperBound=100  
Meine 5. Vermutung ist: 97  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=98, upperBound=100  
Meine 6. Vermutung ist: 99  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=100, upperBound=100  
Die Zahl 100 im 7. Versuch gefunden.

### Testfall #3 (gesucht ist 50)

Bitte denken Sie sich eine Zahl zwischen 0 und 100 aus!  
Meine 1. Vermutung ist: 50  
Bitte gib eine Antwort ein (<, >, =): =  
lowerBound=0, upperBound=100  
Die Zahl 50 im 1. Versuch gefunden.

### Testfall #4 (gesucht ist 22)

Bitte denken Sie sich eine Zahl zwischen 0 und 100 aus!  
Meine 1. Vermutung ist: 50  
Bitte gib eine Antwort ein (<, >, =): <  
lowerBound=0, upperBound=49  
Meine 2. Vermutung ist: 24  
Bitte gib eine Antwort ein (<, >, =): <  
lowerBound=0, upperBound=23  
Meine 3. Vermutung ist: 11  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=12, upperBound=23  
Meine 4. Vermutung ist: 17  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=18, upperBound=23  
Meine 5. Vermutung ist: 20  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=21, upperBound=23  
Meine 6. Vermutung ist: 22  
Bitte gib eine Antwort ein (<, >, =): =  
lowerBound=21, upperBound=23  
Die Zahl 22 im 6. Versuch gefunden.

### Testfall #5 (gesucht ist 21)

Bitte denken Sie sich eine Zahl zwischen 0 und 100 aus!  
Meine 1. Vermutung ist: 50  
Bitte gib eine Antwort ein (<, >, =): <  
lowerBound=0, upperBound=49  
Meine 2. Vermutung ist: 24  
Bitte gib eine Antwort ein (<, >, =): <  
lowerBound=0, upperBound=23  
Meine 3. Vermutung ist: 11  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=12, upperBound=23  
Meine 4. Vermutung ist: 17  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=18, upperBound=23  
Meine 5. Vermutung ist: 20  
Bitte gib eine Antwort ein (<, >, =): >  
lowerBound=21, upperBound=23  
Meine 6. Vermutung ist: 22  
Bitte gib eine Antwort ein (<, >, =): <  
lowerBound=21, upperBound=21  
Die Zahl 21 im 7. Versuch gefunden.