数当てゲーム

これまで学習してきた乱数（第2章）・if 文（第3章）・do 文を応用して、**数当てゲーム**を作りましょう。そのプログラムが **List 4-3** です。

変数 no が《当てるべき数》です。その値は、0 以上 99 以下の乱数として生成します。

List 4-3 Chap04/Kazuate.java

```
// 数当てゲーム（0〜99を当てさせる）

import java.util.Random;
import java.util.Scanner;

class Kazuate {

   public static void main(String[] args) {
      Random rand = new Random();
      Scanner stdIn = new Scanner(System.in);

      int no = rand.nextInt(100);    // 当てるべき数：0〜99の乱数として生成

      System.out.println("数当てゲーム開始!!");
      System.out.println("0〜99の数を当ててください。");

      int x;                         // プレーヤが入力した数
      do {
         System.out.print("いくつかな：");
         x = stdIn.nextInt();                                    ← 1

         if (x > no)
            System.out.println("もっと小さな数だよ。");
         else if (x < no)                                        ← 2
            System.out.println("もっと大きな数だよ。");
      } while (x != no);                                         ← 不正解であれば繰り返す。

      System.out.println("正解です。");
   }
}
```

実行例

```
数当てゲーム開始!!
0〜99の数を当ててください。
いくつかな：50
もっと大きな数だよ。
いくつかな：75
もっと小さな数だよ。
いくつかな：62
正解です。
```

Fig.4-4 に示すフローチャートとプログラムとを見比べながら理解しましょう。

1 『いくつかな：』と数値の入力を促して、変数 x に値を読み込みます。

2 読み込んだ x の値が no より大きければ『もっと小さな数だよ。』と表示し、x の値が no より小さければ『もっと大きな数だよ。』と表示します。

▶ この時点では、x と no の値が等しければ、何も表示しません。

それから do 文を繰り返すかどうかの判定を行います。判定のための制御式は、

```
x != no          // 読み込んだxと当てるべき数noが等しくないか？
```

です。そのため、読み込んだ x の値が当てるべき数 no と**等しくないあいだ** do 文が繰り返されます。

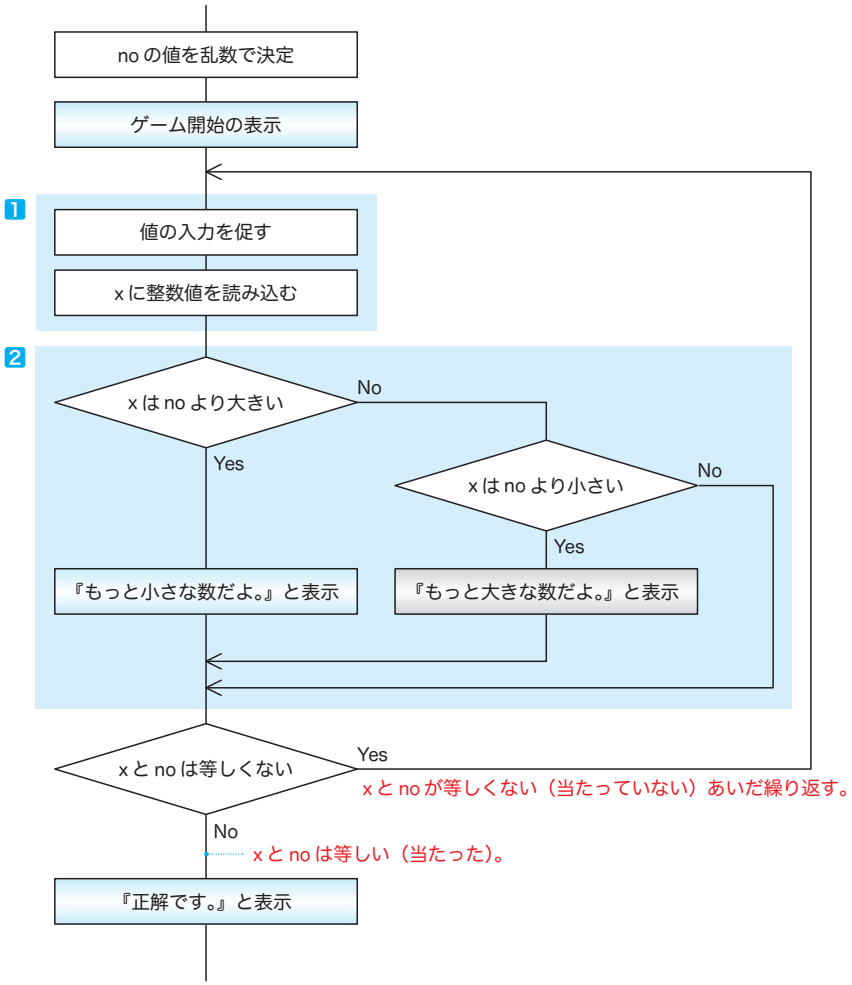

Fig.4-4　List 4–3 のフローチャート

数が当たったら（読み込んだ *x* が、当てるべき数 *no* と等しくなったら）、**do** 文は終了します。『正解です。』と表示してプログラムを終了します。

演習 4–2

２桁の整数値（10 ～ 99）を当てさせる《数当てゲーム》を作成せよ。

演習 4–3

右に示すように、二つの整数値を読み込んで、小さいほうの数以上で大きいほうの数以下の全整数を小さいほうから順に表示するプログラムを作成せよ。

```
整数Ａ：37 ↵
整数Ｂ：28 ↵
28 29 30 31 32 33 34 35 36 37
```