Worked Example 9.1

Investigating Number Sequences

In this example, we investigate properties of number sequences. A number sequence can be a sequence of measurements, prices, random values, or mathematical values (such as the sequence of prime numbers).

There are many interesting properties that can be investigated. For example, you can look for hidden patterns or test whether a sequence is truly random. We will look at a fairly simple question: how the last digit of each value is distributed. For a given sequence, we will produce a chart such as

```
0: 105
1: 94
2: 81
3: 112
4: 89
5: 103
6: 103
7: 100
8: 108
9: 105
```

In order to produce arbitrary sequences, we declare an interface type with a single method:

```java
public interface Sequence
{
    int next();
}
```

The `LastDigitDistribution` class analyzes sequences. It keeps an array of ten counters. Its `process` method receives a `Sequence` object and updates the counters:

```java
public void process(Sequence seq, int valuesToProcess)
{
    for (int i = 1; i <= valuesToProcess; i++)
    {
        int value = seq.next();
        int lastDigit = value % 10;
        counters[lastDigit]++;
    }
}
```

Note that this method has no knowledge how the sequence values are produced.

To analyze a specific sequence, you provide a class that implements the `Sequence` interface. Here are two examples: the sequence of perfect squares (0 1 4 9 16 25 ...), and a sequence of random integers.

```java
public class SquareSequence implements Sequence
{
    private int n;

    public int next()
    {
        n++;
        return n * n;
    }
}
```

```
public class RandomSequence implements Sequence
{
    public int next()
    {
        return (int) (Integer.MAX_VALUE * Math.random());
    }
}
```

The following class demonstrates the analysis process. Note the pattern of the last digits of the sequence of perfect squares.

```
public class SequenceDemo
{
    public static void main(String[] args)
    {
        LastDigitDistribution dist1 = new LastDigitDistribution();
        dist1.process(new SquareSequence(), 1000);
        dist1.display();
        System.out.println();

        LastDigitDistribution dist2 = new LastDigitDistribution();
        dist2.process(new RandomSequence(), 1000);
        dist2.display();
    }
}
```

Program Run

```
0: 100
1: 200
2: 0
3: 0
4: 200
5: 100
6: 200
7: 0
8: 0
9: 200

0: 105
1: 94
2: 81
3: 112
4: 89
5: 103
6: 103
7: 100
8: 108
9: 105
```

The complete program is contained in the ch09/sequence directory of the book's source code.