Figure 6.3 A first look at *loops*.

How could we solve the problem of adding up the numbers from 1 to 5.

- **don't start by writing code!**
- think first of how you would approach the problem yourself
  - one possibility: "5 + 4 is 9, 9 + 3 is 12, 12 + 2 is 14, …"
  - i.e. keep adding the numbers to a "running total"

In Java, if **sum** is the running total and **number** is the current number,

sum = sum + number;

would do this. And we could start with

int number = 5;

But we have to also change **number** to **4, 3, 2, 1,** and then somehow stop when we're done.

This calls for a *loop* construct: **while (** *something* **) { … }**.

The *loop body* **{ … }**

- is executed over and over again,
- for as long as *something* is **true**

In our case the loop body will look like

```
sum = sum + number;
number = number – 1;
```

and we should continue for as long as **number** is one of **5, 4, 3, 2, 1**; i.e. for as long as **number > 0**.

So, with appropriate initialization, our loop will look like:

```
int number = 5;    // the first number to be accumulated
int sum = 0;       // so far, we've added no numbers at all

while (number > 0) {
    sum = sum + number;
    number = number – 1;
}
```

Note: **number > 0** is another example of a **boolean** expression.