do-while Loop

This lesson explains how do-while loop is implemented and how it is different from the while loop.

In a previous lesson, we saw the steps in which the `while` loop is executed:

```
preparation

condition check
actual work
iteration

condition check
actual work
iteration
...
```

The `do-while` loop is very similar to the `while` loop. The difference is that the condition check is performed at the end of each iteration of the `do-while` loop so that the actual work is performed at least once:

```
preparation

actual work
iteration
condition check  ← at the end of the iteration

actual work
iteration
condition check  ← at the end of the iteration
...
```

For example, `do-while` may be more natural in the following program, where the user guesses a number. The user must guess at least once so that the number can be compared:

```
import std.stdio;
```

```
import std.random;

void main() {

    int number = uniform(1, 101);

    writeln("I am thinking of a number between 1 and 100.");

    int guess;

    do {
        write("What is your guess? ");

        readf(" %s", &guess);

        if (number < guess) {
            write("My number is less than that. ");

        } else if (number > guess) {
            write("My number is greater than that. ");
        }

    } while (guess != number);

    writeln("Correct!");
}
```

Use of do-while loop

The function `uniform()` that is used in the program is a part of the `std.random` module. It returns a random number in the specified range. In the way it is used above, the second number (101) is considered to be outside of the range. In other words, `uniform()` would not return 101 for that call.

In the next lesson, you will find a quiz to test the concepts covered in this chapter.