

Statements

Actions associated with events, transitions, process flowchart objects, agents, controls, etc. in AnyLogic are written in Java code. Java code is composed of statements. *Statement* is a unit of code, an instruction provided to a computer. Statements are executed sequentially one after another and, generally, in top-down order. The following code of three statements, for example, declares a variable `x`, assigns it a random number drawn from a uniform distribution, and prints the value to the model log.

```
double x;  
  
x = uniform();  
  
traceln( "x = " + x );
```

You may have noticed that there is a semicolon at the end of each statement. In Java this is a rule:

Each statement in Java must end with semicolon except for the block `{...}`.

We will consider these types of Java statements:

- Variable declaration, e.g.: `String s;` or `double x = getX();`
- Function call: `traceln("Time:" + time());`
- Assignment: `shipTo = client.address;`
- Decisions: `if(...) then { ... } else { ... },` `switch(...) { case ... : case ...; ... }`
- Loops: `for() { ... }, while(...) { ... }, do { ... } while(...)` and **also break;** and **continue;**
- Block: `{ ... }`

It is important that, regardless of its computational complexity and the required CPU time, any piece of Java code written by the user is treated by AnyLogic simulation engine as indivisible and is executed logically instantly: the model clock is not advanced.