Loops

While, for, do-while

Keyword break and continue

Problem

Print all the numbers

- From 1 to 5
- From 1 to 1000
- From 1 to n
- From n to m

What is a loop?

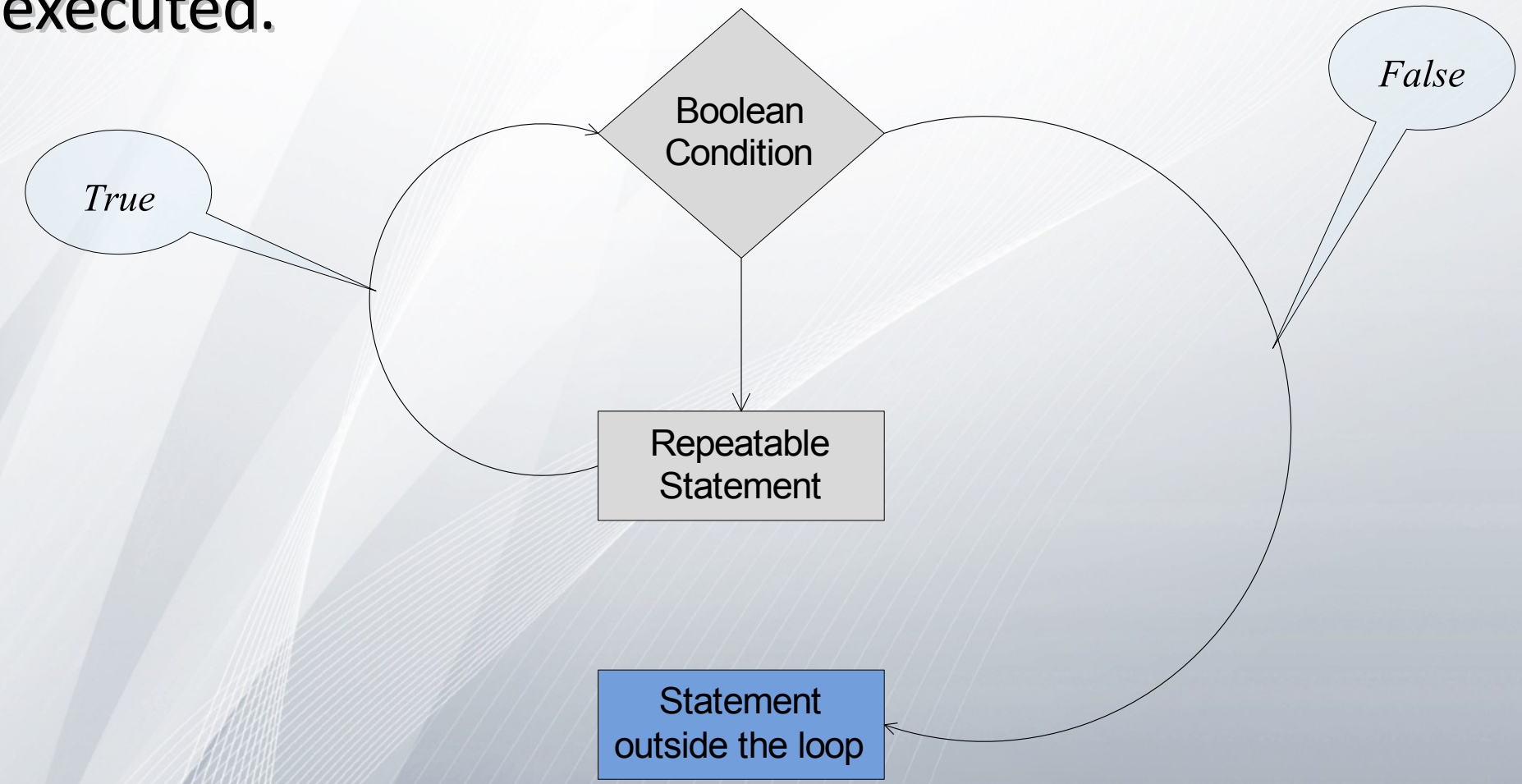
- A loop is a structure that allows sequence of statement to be executed more times in a row

- Loops have a boolean condition and a block of code for execution. While the condition is true, the block is being executed.

- A loop that never ends is called an infinite loop

While

- While the condition is true, the block is being executed.



While

Counter initialization

Boolean condition.

If i > 100, the next block will be skipped

```
int i = 1;

while (i <= 100) {
    System.out.println(i);
    i++;
}
```

Block of code

repeatable execution

Demo

- How to print all numbers between 2 and 50?
- How to print all **even** numbers between 2 and 50
- How to do it with less iterations?

Excercises

- Print all numbers between 1 and n, which can be devided by 7 without a reminder
- Modify the task to print the numbers in reverse order (n to 1)
- We need to enter a number and the program needs to print the following:

```
Enter n: 3
1 2 3
2 4 6
3 6 9
```

```
Enter n: 4
1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16
```

do-while

- Similar to while loop

  but
- always enters the execution at least once

  because
- Condition is after the execution

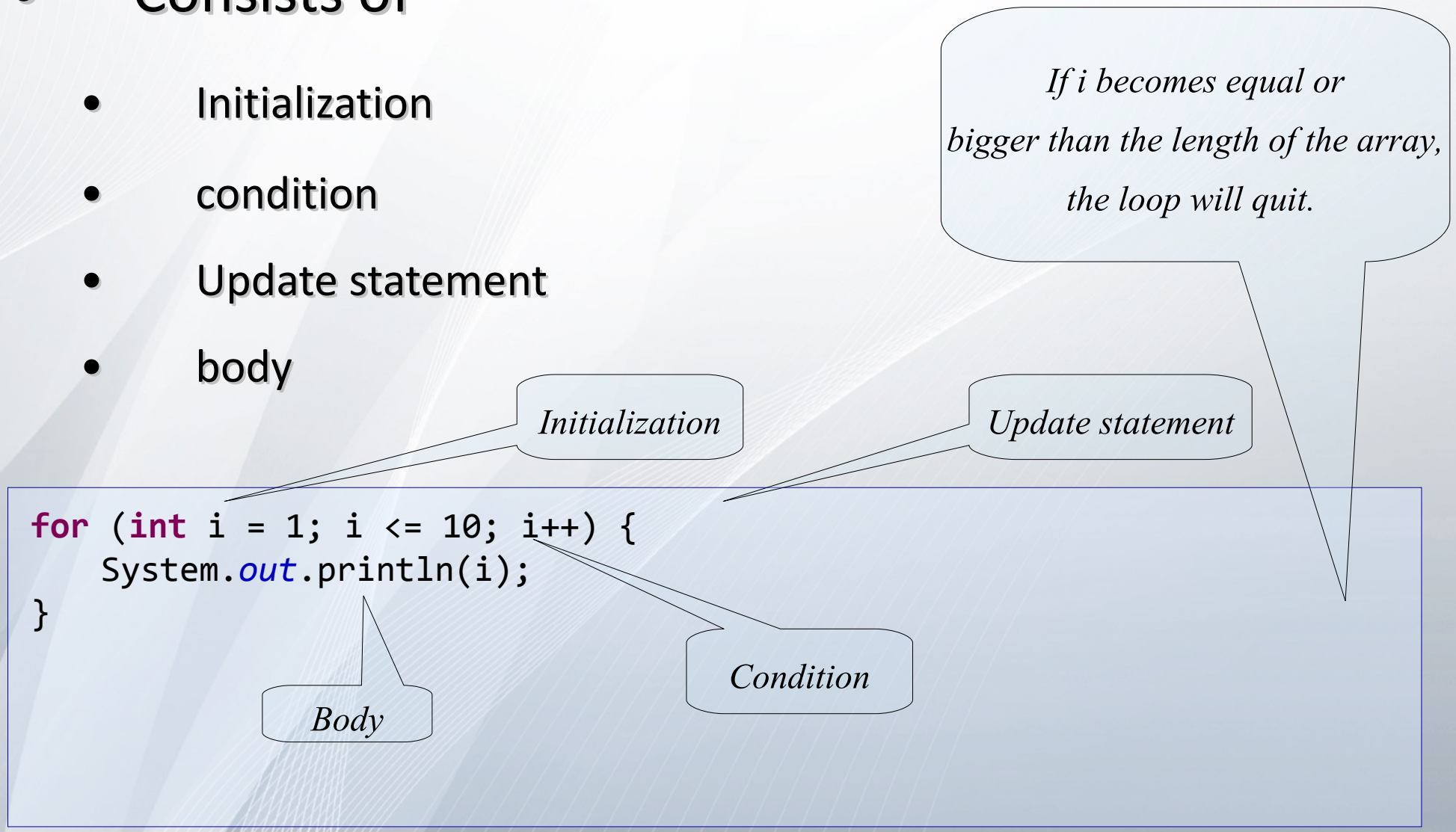
```
do {
    System.out.println(i);
    i++;
} while (i <= 1000)
```

Execute the block of code

Check if i<=1000. If it's true, repeat once more.

For loop

- Consists of
  - Initialization
  - condition
  - Update statement
  - body



Exercise

- Enter an integer. For example, if you enter 3, you shoud see the following:

  „3 3 3 !“

- If you enter 5, you should see:

  „5 5 5 5 5 !“

Problem

- Try to quit a for-loop during the execution of the repeatable block
- One possible to solution is to set the counter to a value which will make the boolean condition quit the loop....but

Break

- Break is a keyword
- A statement by itself
- It doesn't require anything else
- It stops the execution of the loop

The loop will quit when i is equal to 7

```
for (int i = 0; i < 50; i++) {
    if (i == 7) {
        break;
    }
}
```

Problem

- Try to omit specific block of code in the body – for example sum all numbers between 1 and 100 but omit all numbers between 51 and 74
- Encapsulating the code in if-else statements may be used. Although for more complicated structures should be used for more complicated cases

Continue

- Continue is a keyword
- A statement by itself
- It doesn't require anything else
- It stops the current iteration of the loop, but doesn't stop the loop

```
for (int i = 0; i < 100; i++) {
    if (i > 51 && i < 74) {
        continue;
    }
    sum = sum + i;
}
```

If i is between 51 na 74,

the loop will skip

all statements after ***continue.***

Exercise

- Calculate factorial of a number, where the numer is entered by the user
- Calculate the sum of the first n numers of the Fibonacci sequence:

  0 + 1 + 1 + 2 + 3 + 5 + 8 + ...

- Calculate the following for user specified n:

  $(1!) / (1^1) + (2!) / (2^2) + ... + (n!) / (n^n)$

Summary

- Why do we use loops?
- What does a loop consist of?
- Difference between *while* and *do-while*?
- How to use *for* – loop?
- How to terminate a loop?
- How to stop the current iteration?