LOOPING STATEMENTS

OVERVIEW

- There may be situation when you need to execute a block of code several number of times
- A loop statement allows us to execute a statement or group of statements multiple times
- Looping statements available:

  1. while
  2. for
  3. do…while

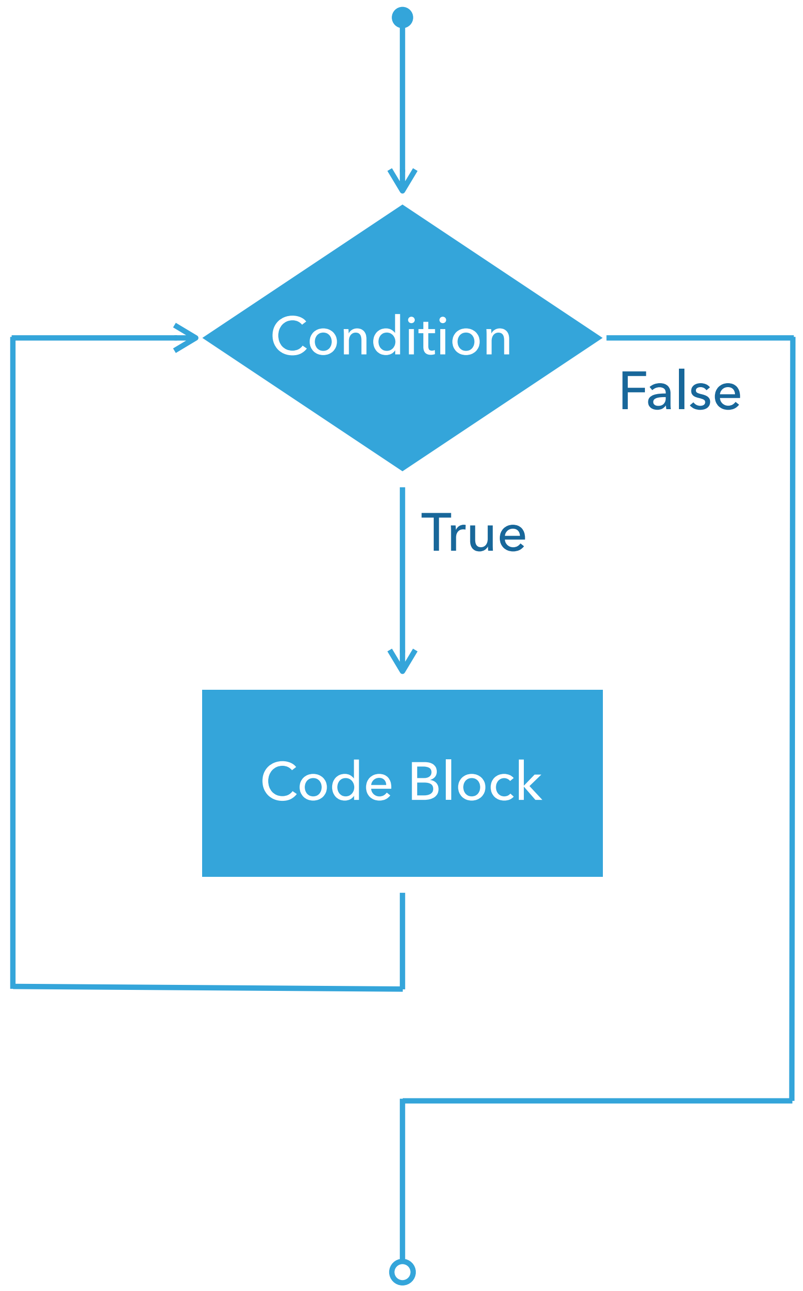
COMMON LOOPS STRUCTURE

- There is a control variable, called the loop counter
- Loop variable must be initialized
- The increment or decrement of the control variable, which is modified each time the iteration of the loop occurs
- The loop condition that determines if the looping should continue or the program should break from it

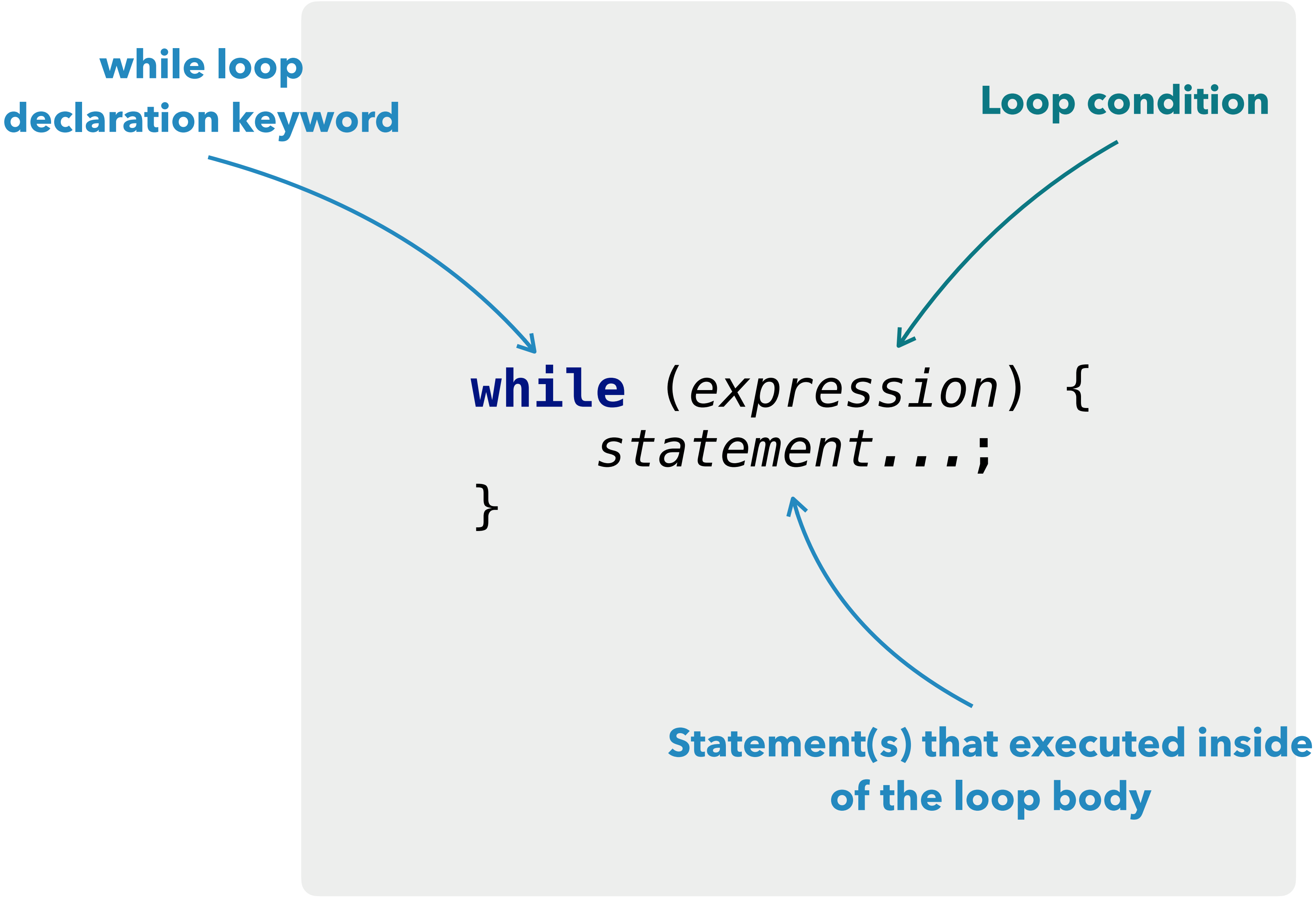
WHILE LOOP: SUMMARY

- Repeats a statement or block of statements while its controlling boolean expression is true
- Boolean expression is evaluated before the first iteration of the loop, hence executed zero or many times
- Usually used when number of iterations depends

WHILE LOOP: FLOWCHART



WHILE LOOP: SYNTAX



while loop declaration keyword

Loop condition

```
while (expression) {
    statement...;
}
```

Statement(s) that executed inside of the loop body

WHILE LOOP: CODE EXAMPLE

Code

```java
int i = 0;
while (i < 5) {
    System.out.print("i = " + i + "; ");
    i++;
}
```

Console output

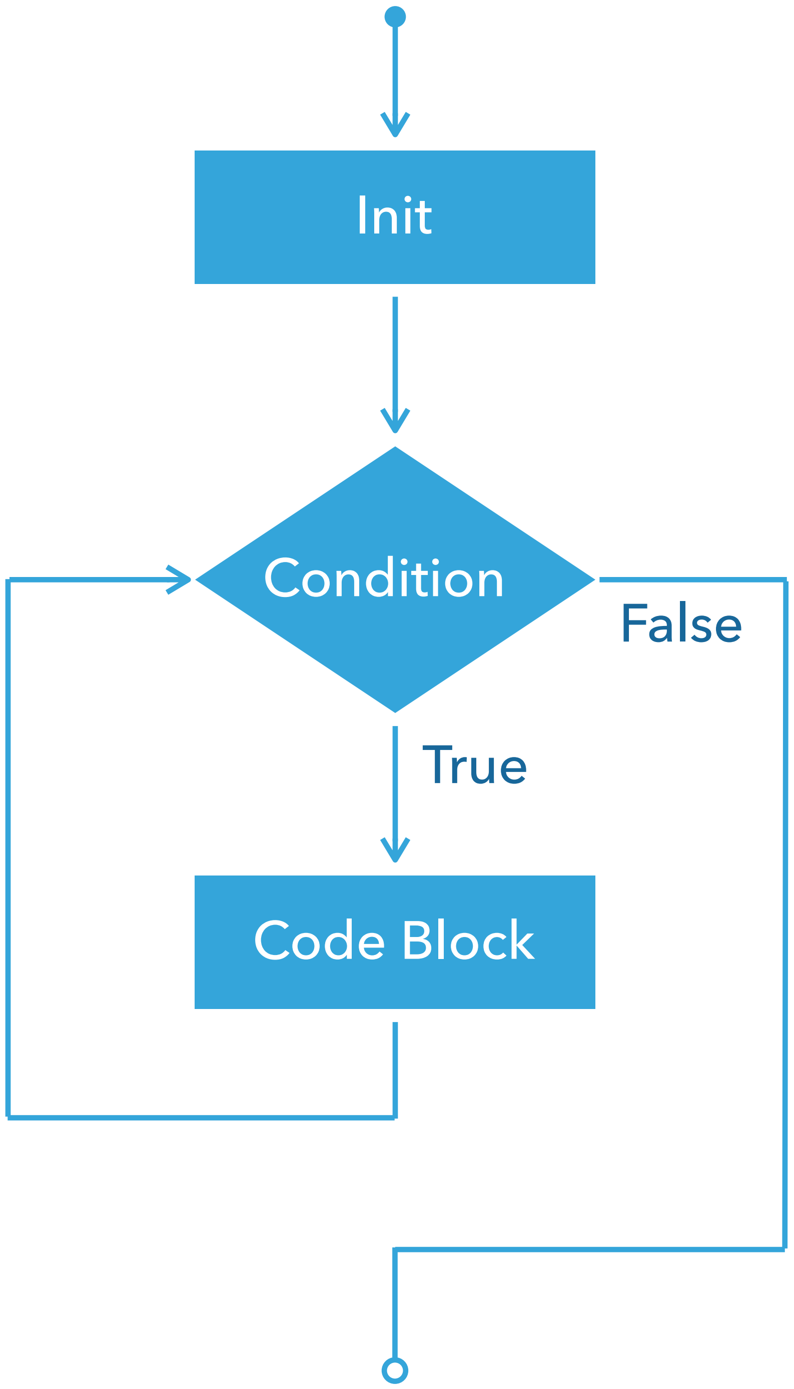
```
i = 0; i = 1; i = 2; i = 3; i = 4;

Process finished with exit code 0
```

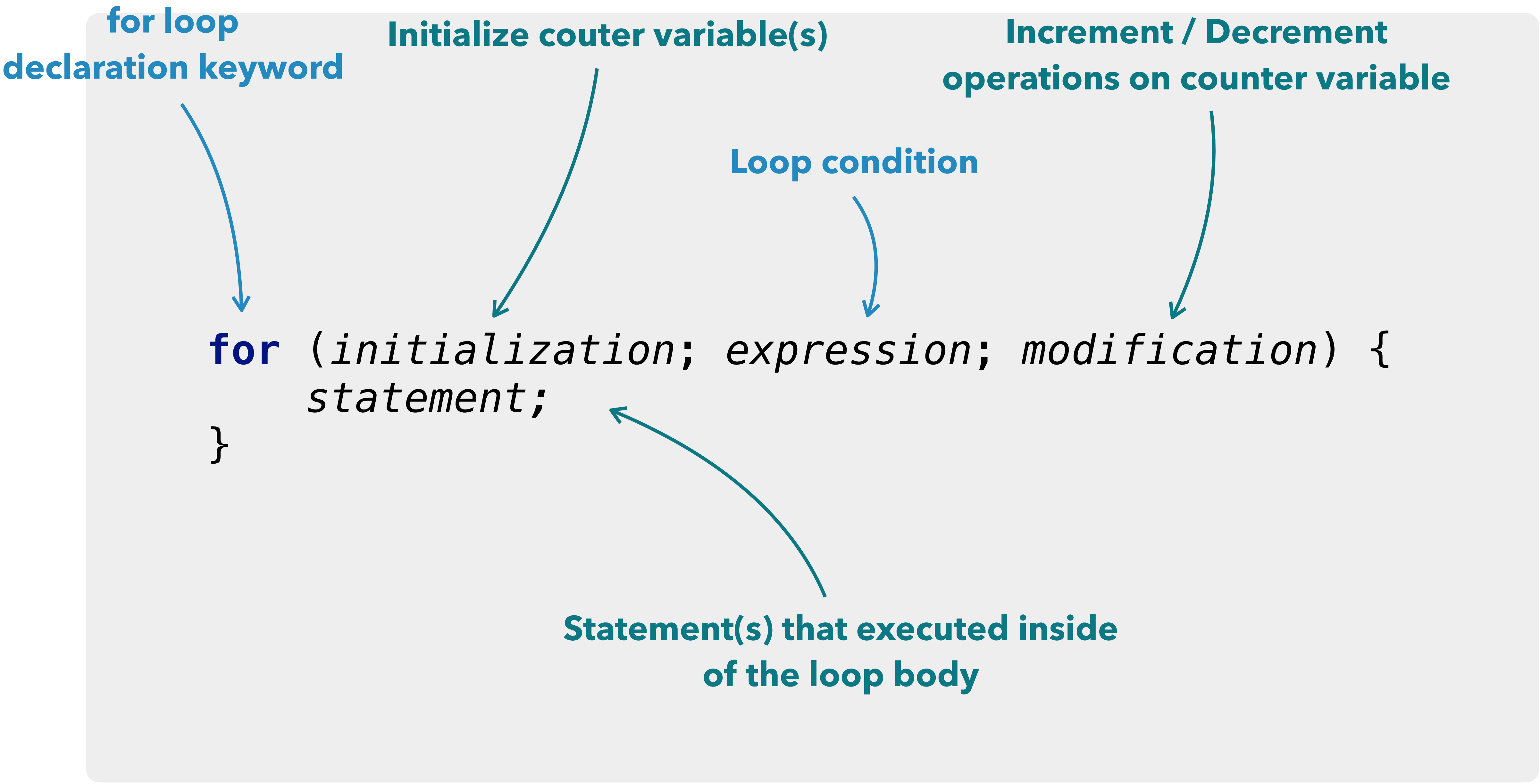
FOR LOOP: SUMMARY

- Control structure that allows us to repeat certain operations by incrementing or decrementing and evaluating a loop counter
- Boolean expression is evaluated before the first iteration of the loop, hence executed zero or many times
- Usually used when number of iterations are known in advance

FOR LOOP: FLOWCHART



FOR LOOP: SYNTAX



for loop declaration keyword

Initialize couter variable(s)

Increment / Decrement operations on counter variable

Loop condition

```
for (initialization; expression; modification) {
    statement;
}
```

Statement(s) that executed inside of the loop body

FOR LOOP: CODE EXAMPLE

Code

```java
for (int i = 0; i < 5; i++) {
    System.out.print("i = " + i + "; ");
}
```

Console output

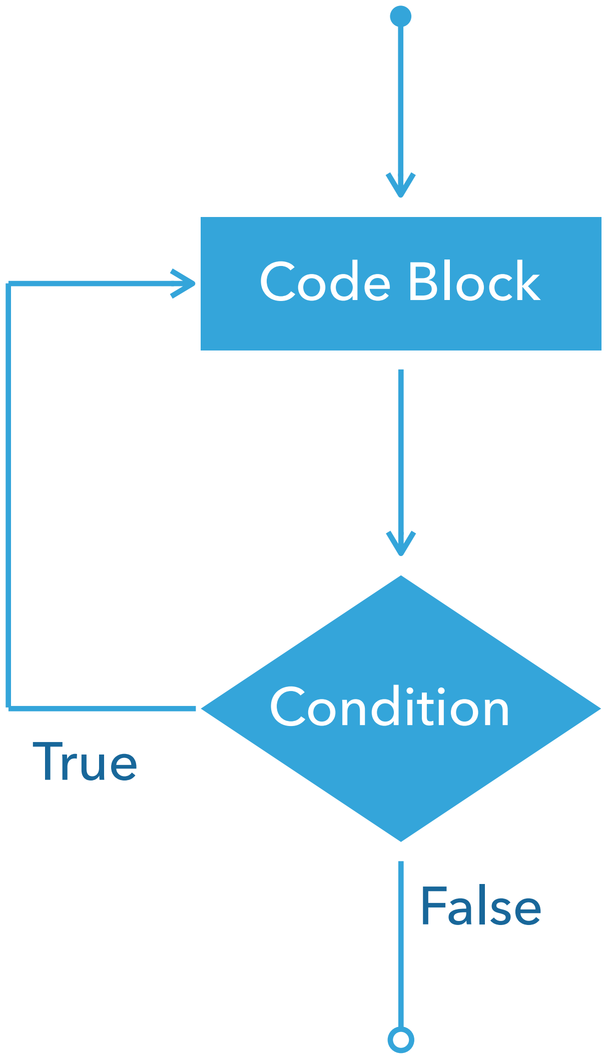
```
i = 0; i = 1; i = 2; i = 3; i = 4;

Process finished with exit code 0
```

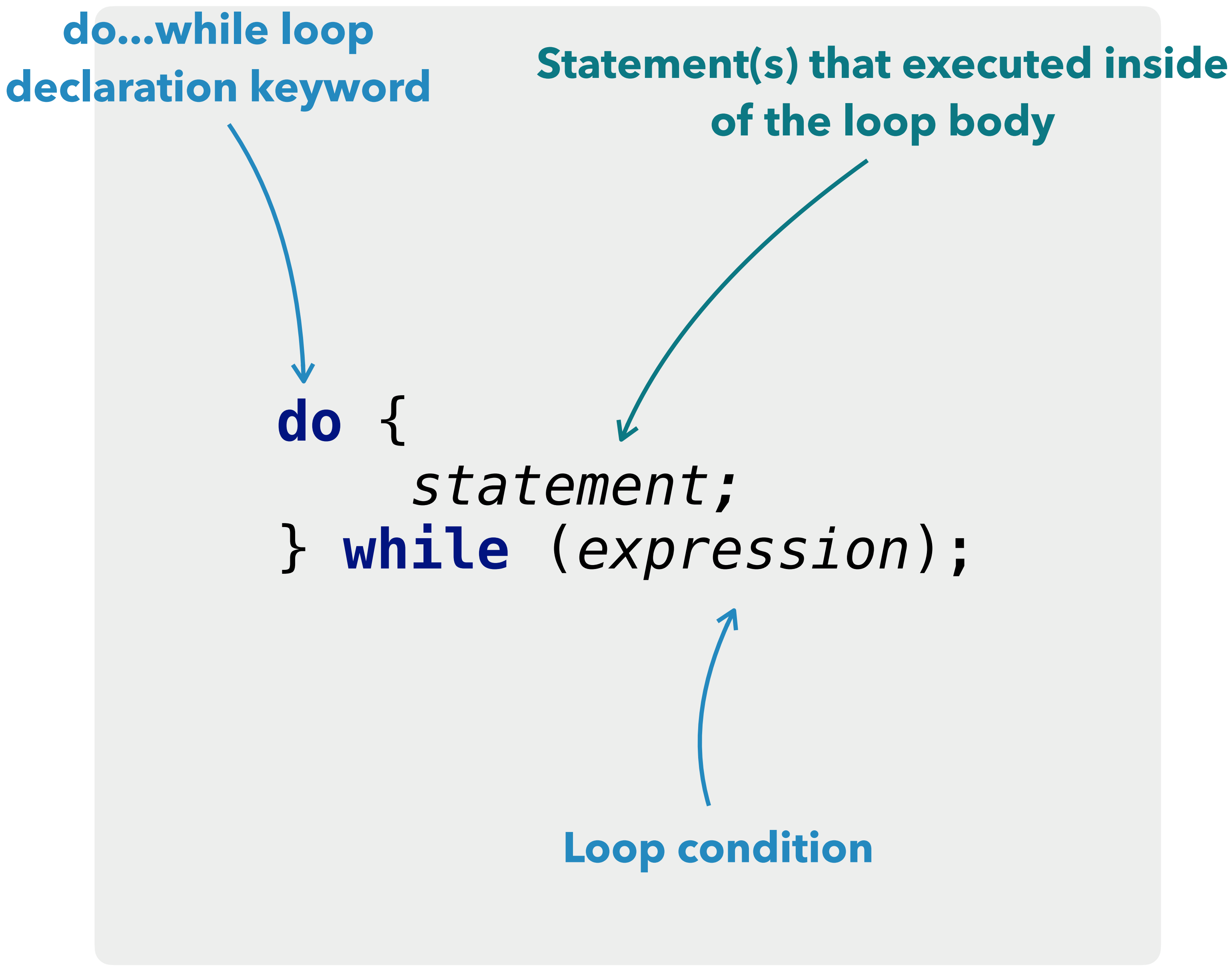
DO WHILE LOOP: SUMMARY

- Repeats a statement or block of statements while its controlling boolean expression is true
- Boolean expression is evaluated after the first iteration of the loop, hence executed one or many times
- Usually used when number of iterations are known in advance

DO WHILE LOOP: FLOWCHART



DO WHILE LOOP: SYNTAX



do...while loop declaration keyword

Statement(s) that executed inside of the loop body

```
do {
      statement;
} while (expression);
```

Loop condition

DO WHILE LOOP: CODE EXAMPLE

Code

```
int i = 0;
do {
    System.out.print("i = " + i + "; ");
    i++;
} while (i < 5);
```

Console output

```
i = 0; i = 1; i = 2; i = 3; i = 4;

Process finished with exit code 0
```

BRANCHING STATEMENTS IN LOOPS

BRANCHING STATEMENTS IN LOOPS

- Branching statements are used to change normal flow of execution based on some condition
- Branching statements available in loops:
  1. break
  2. continue

BREAK STATEMENT: OVERVIEW

- Terminates the innermost for, while, do…while statement
- When the break statement encountered, the loop is immediately terminated and the program control resumes at the next statement following the loop

BREAK STATEMENT: EXAMPLE

Code

```
for (int i = 0; i < 10; i++) {
    if (i == 3) {
        break;
    }
    System.out.print("i = " + i + "; ");
}
```

Console output

```
i = 0; i = 1; i = 2;

Process finished with exit code 0
```

CONTINUE STATEMENT: OVERVIEW

- In a for loop, the continue keyword causes control to immediately jump to the modification statement
- In a while or do…while loop, causes control to immediately jump to the boolean expression

CONTINUE STATEMENT: EXAMPLE

Code

```
for (int i = 0; i < 10; i++) {
    if (i % 2 == 0) {
        continue;
    }
    System.out.print("i = " + i + "; ");
}
```

Console output

```
i = 1; i = 3; i = 5; i = 7; i = 9;

Process finished with exit code 0
```

REFERENCES

- https://www.tutorialspoint.com/java/java_loop_control.htm
- https://www.baeldung.com/java-loops
- https://www.developer.com/java/data/using-different-types-of-java-loops-looping-in-java.html
- https://docs.oracle.com/javase/tutorial/java/nutsandbolts/branch.html