

# Loops and Arrays

---

Tobias Hanf, Maik Göken

November 7, 2022

Learn Programming with Java

# Outline

Very Important Quote for Today

What is a Loop

Different kinds of loops

Arrays

Exercises

Very Important Quote for Today

---

*There are 2 hard problems in computer science: cache invalidation, naming things, and off-by-1 errors.*

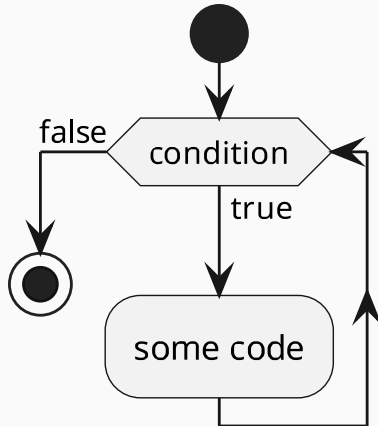
Source: <https://twitter.com/secretGeek/status/7269997868>

# What is a Loop

---

A Loop is a programming concept which allows the **repeated** execution of code until a specific **condition** is reached.

# Loop Diagram



## Different kinds of loops

---



# For Loop

```
1 // loop header
2 for(<initial value>; <condition>; <expression>) {
3     // loop body
4     <code to be repeated>
5
6 }
```

- <initial value> : declare and initialize a **counter**
- <condition> : loop runs until this is **false**
- <expression> : increment the **counter**

## For Loop (Example)

```
1 public class ForExample {  
2  
3     public static void main(String[] args) {  
4         for(int i = 0; i <= 10; i++) {  
5             System.out.print("na ");  
6         }  
7         System.out.println("BATMAN!");  
8     }  
9  
10 }
```

# While Loop

```
1 while(<condition>) {  
2  
3   <code to be repeated>  
4  
5 }
```

- `<condition>`: loop runs until this is false

## While Loop (Example)

```
1 public class WhileExample {  
2  
3     public static void main(String[] args) {  
4         int a = 0;  
5         while(a <= 10) {  
6             System.out.println(a);  
7             // Otherwise you would get an endless loop  
8             a++;  
9         }  
10    }  
11  
12 }
```

# Do-While Loop

```
1 do {  
2  
3   <code to be repeated>  
4  
5 } while(<condition>)
```

- `<condition>`: loop runs until this is `false`
- Executes the code `first`
- Only checks the condition afterwards

## Do-While Loop (Example)

```
1 public class DoWhileExample {  
2  
3     public static void main(String[] args) {  
4         int a = 0;  
5         do {  
6             System.out.println(a);  
7             // Otherwise you would get an endless loop  
8             a++;  
9         } while(a <= 10)  
10    }  
11  
12 }
```

## Useful statements

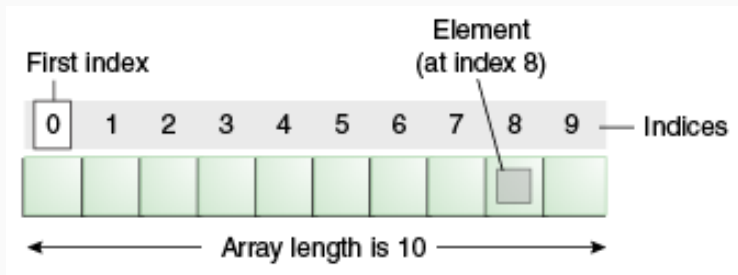
- `break` will exit the loop
- `continue` will exit the current iteration (jump to header)

# Arrays

---



# What are arrays



- **Fixed length** block of memory
- Can store **multiple** elements of **one** datatype
- Elements can be accessed via the `[]` operator
- Elements can be accessed **randomly**
- Java arrays are objects

Image source: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>

# Code

```
1 <type>[] <name>; //declaration
2 <name> = new <type>[<size>]; //allocation
3 <name>[<index>] ; // access
4 <name>.length; // length of array
```

```
1 int[] arr; // declare array
2 arr = new int[10]; //allocate memory for 10 elements
3
4 arr[0] = 10; //store 10 into first element
5
6 arr[2] = 5;
7
8 System.out.println(arr[2]); // what gets printed ?
```

## References

```
1 int[] a;  
2 int[] b;  
3  
4 a = new int[10];  
5 a[0] = 22;  
6 b = a;  
7  
8 System.out.println(b[0]); // ?  
9  
10 b[0] = 11;  
11  
12 System.out.println(b[0]); // ?  
13 System.out.println(a[0]); // ?
```

## References

- `int[] a` is only a **reference** to memory
- Assigning an array only copies the reference, **not** the memory
- To copy an array use `System.arraycopy(...)`
- Or write your own copy function (good exercise)

# Exercises

---

Print all **even** numbers from 1 to n.

# Staircase

## Description

Print a staircase of \* starting with one \*. Each step of the staircase should be one \* longer than its predecessor.

## Example Output (n=5)

```
*  
**  
***  
****  
*****
```

## Array reversal

Create an array of size 10 and fill it with data. Print each element of this array, first in normal order then in reverse order.



Implement (or copy) FizzBuzz and let it run  $n$  times.

## Convert an Integer to String

Convert an **Integer** to a **String** by extracting each **digit** from the Integer, calculating the correct **character** and adding it to a String in the **correct** order.

Usefull opearitions: `\`, `%`, `+`