Java Codecollection

Hello World

```
public class HelloWorld {
	public static void main (String[] args){
		// Ausgabe Hello World!
		System.out.println("Hello World!");
	}
}
```

HashMap

similar to dict in python

- import

```
	import java.util.HashMap;
```

- Initialize

```
	HashMap<String, Integer> hashmap = new HashMap<String, Integer>();
```

- add a key-value-pair

```
	hashmap.put("hinatazaka", 46);
	hashmap.put("AKB", 48);
```

- check if key exists

```
	hashmap.containsKey("keyakizaka");
```

- remove a key-value-pair

```
	int returned_value = hashmap.remove("AKB");
```

- size of a HashMap

```
hashmap.size();
```

List

ArrayList

similar to list in python

- Initialize

```
    ArrayList<Integer> array_list = new ArrayList<Integer>();
```

- add an element

```
    array_list.add(42);
    array_list.add(46);
```

- get an element

```
    array_list.get(0);
```

- remove an element

```
    array_list.remove(0);
```

- check whether an element is in the arraylist

```
    boolean exits = array_list.contains(46);
```

- size or length of an arraylist

```
array_list.size();
```

- get and remove the element at `index` from an arraylist

```
array_list.remove(int index);
```

- find min of an arraylist

```
minimum = Collections.min(array_list);
```

Stack

- initiate

```
Stack<int> stack = new Stack<>();
```

- push (Pushes an item onto the top of this stack.)

```
stack.push(5);
```

- pop (return the object at the top of this stack)

```
stack.pop();
```

- peek (return the object at the top of this stack)

```
stack.peek();
```

- empty (`true` if stack is empty, `false` otherwise)

```
stack.empty();
```

- search (return the 1-based position from the top of the stack where the object is located; the return value `-1` indicates that the object is not on the stack.)

```
stack.search(7);
```

For

iterate elements in list

```
      for(int n: array_list){
            System.out.println(n);
```

```
        }
```