

# Objektorientierte Programmierung: Aufgabenblatt 4

Florian Ludewig (185722)

5. Juni 2020

## Aufgabe 1

### Java Implementierung

Einwohner.java

```
1 public class Einwohner {
2     protected double steuerFaktor = 0.1;
3     protected int einkommen;
4     protected int mindestAbgabe = 1;
5
6     int zuVersteuerndesEinkommen() {
7         return this.einkommen;
8     }
9
10    int steuer() {
11        int abgabe = (int) Math.floor(this.zuVersteuerndesEinkommen() * this.
12            steuerFaktor);
13        return abgabe > this.mindestAbgabe ? abgabe : this.mindestAbgabe;
14    }
15
16    void setEinkommen(int einkommen) {
17        this.einkommen = einkommen;
18    }
}
```

Adel.java

```
1 public class Adel extends Einwohner {
2     Adel() {
3         this.mindestAbgabe = 20;
4     }
5 }
```

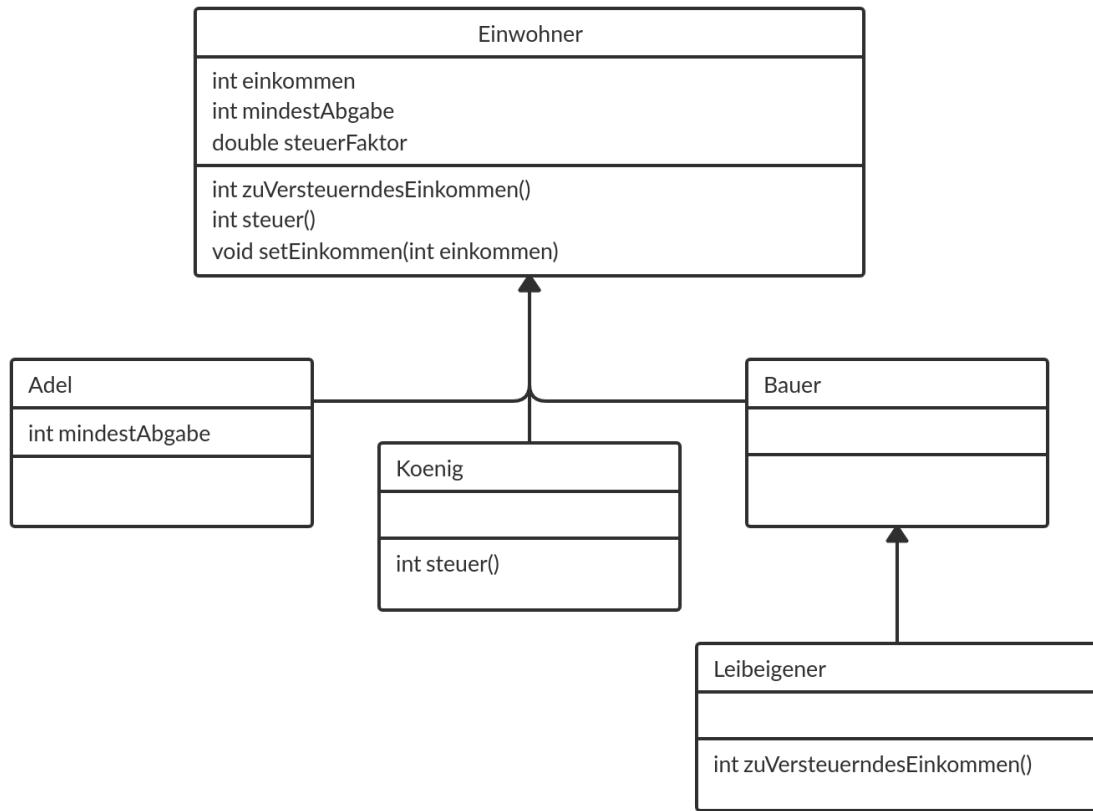
Koenig.java

```
1 public class Koenig extends Einwohner {
2     int steuer() {
3         /*
4             * Ich gehe davon aus, dass der Koenig auch den Mindestbetrag von 1
5             * Gulden nicht
6             * zahlen muss (ist aus 3. nicht genau hervorgegangen)
7             */
8         return 0;
9     }
}
```

Leibeigener.java

```
1 public class Leibeigener extends Bauer {
2     int zuVersteuerndesEinkommen() {
3         return this.einkommen > 12 ? this.einkommen - 12 : 12;
4     }
5 }
```

## Ergänztes Klassendiagramm



## Aufgabe 2

### Array

DIAarray.java

```
1 import java.util.Arrays;
2
3 public class DIAarray extends DynIntArray {
4     int array[] = new int[0];
5
6     void add(int e) {
7         int newArray[] = new int[this.array.length + 1];
8         for (int i = 0; i < this.array.length; i++) {
9             newArray[i] = this.array[i];
10        }
11        newArray[this.array.length] = e;
12        this.array = newArray;
13    }
14
15    void setElementAt(int i, int e) {
16        this.array[i] = e;
17    }
18
19    int getElementAt(int i) {
20        return this.array.length > i ? this.array[i] : 0;
21    }
22
23    int getElementCount() {
24        return this.array.length;
25    }
26
27    void print() {
28        System.out.println(Arrays.toString(this.array));
29    }
30 }
```

### List

Node.java

```
1 public class Node {
2     Node previous;
3     Node next;
4     int value = 0;
5
6     Node() {}
7     Node(int value) {
8         this.value = value;
9     }
10 }
```

```

DIList.java

1 public class DIAList extends DynIntArray {
2     Node firstNode;
3
4     void add(int e) {
5         Node last = this.getLastNode();
6         Node inserted = new Node(e);
7         if (last == null) this.firstNode = inserted;
8         else last.next = inserted;
9     }
10
11    void setElementAt(int i, int e) {
12        Node node = this.getNodeAt(i);
13        if (node != null) node.value = e;
14    }
15
16    int getElementAt(int i) {
17        Node node = this.getNodeAt(i);
18        return node == null ? 0 : node.value;
19    }
20
21    int getElementCount() {
22        Node temp = this.firstNode;
23        int i = 1;
24        while (temp.next != null) {
25            temp = temp.next;
26            i++;
27        }
28        return i;
29    }
30
31    void print() {
32        String out = "[";
33        Node temp = this.firstNode;
34        boolean first = true;
35        while (temp != null) {
36            if (first) {
37                out += temp.value;
38                first = false;
39            } else out += ", " + temp.value;
40            temp = temp.next;
41        }
42        System.out.println(out + "]");
43    }
44
45    private Node getNodeAt(int i) {
46        Node temp = this.firstNode;
47        for (int j = 0; j < i; j++) {
48            if (temp.next == null) return null;
49            temp = temp.next;
50        }
51        return temp;
52    }
53
54    private Node getLastNode() {
55        Node temp = this.firstNode;
56        if (temp == null) return null;
57        while (temp.next != null) {
58            temp = temp.next;
59        }
60        return temp;
61    }
62 }

```