Softwaretechnologie, Übung 2

Henry Haustein

Aufgabe 1

Wir erweitern die Klasse `Book` wie folgt

```
1  private boolean isLent;
2
3  public boolean getLentStatus() {
4    return isLent;
5  }
6
7  public void setLentStatus(boolean status) {
8    this.isLent = status;
9  }
```

Die Klasse `Library` wird wie folgt erweitert

```
1  public Book search(String title) {
2    for (int i = 0; i < counter; i++) {
3      if (title.equals(myBooks[i].getTitle())) {
4        System.out.println("The book with the title " + myBooks[i].
             getTitle() + " exists in the library!");
5        return myBooks[i];
6      }
7    }
8    System.out.println("The book with the title " + title + " does
         not exist in the library!");
9    return null;
10 }
11
12 public void loan(String title) {
13   Book b = search(title);
14   if (b.getLentStatus()) {
15     System.out.println("Buch ist ausgeliehen");
16   }
17   else {
18     b.setLentStatus(true);
19     System.out.println("Buch wurde ausgeliehen");
20   }
21 }
```

Aufgabe 2

Die Klasse `Inhabitant` sieht wie folgt aus

```
1  public class Inhabitant {
2    protected int income;
3
4    public Inhabitant() {
5      this.income = 0;
6    }
7
8    public int getIncome() {
9      return income;
10   }
11
12   public void setIncome(int inc) {
13     if (inc < 0) {
14       throw new IllegalArgumentException("Income negative");
15     }
16     else {
17       this.income = inc;
18     }
19   }
20
21   public int taxableIncome() {
22     return income;
23   }
24
25   public int tax() {
26     return (int) Math.floor(Math.max(1,this.taxableIncome() * 0.1)
           );
27   }
28 }
```

Die Klasse `Noble`

```
1  public class Noble extends Inhabitant {
2    public Noble() {
3      super();
4    }
5
6    public int tax() {
7      return (int) Math.floor(Math.max(20,this.taxableIncome() *
           0.1));
8    }
9  }
```

Die Klasse `King`

```
1  public class King extends Inhabitant {
2    public King() {
3      super();
4    }
5
```

```
6     public int tax() {
7       return 0;
8     }
9   }
```

Die Klasse `Peasant`

```
1   public class Peasant extends Inhabitant {
2     public Peasant() {
3       super();
4     }
5   }
```

Die Klasse `Serf`

```
1   public class Serf extends Peasant {
2     public Serf() {
3       super();
4     }
5
6     public int taxableIncome() {
7       return Math.max(0,this.getIncome() - 12);
8     }
9   }
```

Die Klasse `Koenigreich`

```
1   public class Koenigreich {
2     public void steuerbescheid(Inhabitant i) {
3       int taxIncome = i.taxableIncome();
4       int t = i.tax();
5       System.out.println("Einwohner " + i + " hat Einkommen " + i.
            getIncome() + ", zahlt Steuern auf " + taxIncome + " und
            zahlt Steuern " + t);
6   }
7
8     public void main(String[] args) {
9       Noble n = new Noble();
10      King k = new King();
11      Peasant p = new Peasant();
12      Serf s = new Serf();
13      Serf s2 = new Serf();
14
15      n.setIncome(2000);
16      k.setIncome(3000);
17      p.setIncome(100);
18      s.setIncome(1);
19      s2.setIncome(20);
20
21      steuerbescheid(n);
22      steuerbescheid(k);
23      steuerbescheid(p);
24      steuerbescheid(s);
25      steuerbescheid(s2);
```

```
26     }
27 }
```