

# INLOOP Softwaretechnologie, Generics (Bottle)

HENRY HAUSTEIN

## vollständiger Code

Datei Bottle.java

```
1  public class Bottle<T extends Drink> {
2      private T content;
3
4      public Bottle() {
5          this.content = null;
6      }
7
8      public boolean isEmpty() {
9          if (content == null) {
10             return true;
11          }
12          else {
13             return false;
14          }
15      }
16
17      public void fill(T con) {
18          if (this.isEmpty()) {
19             this.content = con;
20          }
21          else {
22             throw new IllegalStateException("Bottle not empty!");
23          }
24      }
25
26      public T empty() {
27          if (this.isEmpty()) {
28             throw new IllegalStateException("Bottle must be filled");
29          }
30          else {
31             T oldcontent = content;
32             content = null;
33             return oldcontent;
34          }
35      }
36  }
```

Datei Drink.java

```
1 public abstract class Drink {  
2  
3 }
```

Datei Beer.java

```
1 public class Beer extends Drink {  
2     private String brewery;  
3  
4     public Beer(String brew) {  
5         this.brewery = brew;  
6     }  
7  
8     public String getBrewery() {  
9         return brewery;  
10    }  
11  
12    public String toString() {  
13        return brewery;  
14    }  
15 }
```

Datei Wine.java

```
1 public abstract class Wine extends Drink {  
2     private String region;  
3  
4     public Wine(String reg) {  
5         this.region = reg;  
6     }  
7  
8     public String getRegion() {  
9         return region;  
10    }  
11  
12    public String toString() {  
13        return region;  
14    }  
15 }
```

Datei WhiteWine.java

```
1 public class WhiteWine extends Wine {  
2     public WhiteWine(String reg) {  
3         super(reg);  
4     }  
5 }
```

Datei RedWine.java

```
1 public class RedWine extends Wine {  
2     public RedWine(String reg) {
```

```
3      super(reg);  
4    }  
5 }
```

## Erklärung

Die Klassen der Drinks sind selbsterklärend.

In der Klasse `Bottle` steht der Parameter `T` für einen Typ (bzw. Klasse) von Drink. `T` wird damit zum Datentyp und kann damit Werte wie `Beer`, `WhiteWine`, `RedWine`, ... annehmen, sodass im konkreten Fall statt `private T content;` eben `private Beer content;` steht. So zieht sich das dann durch die ganze Klasse.