Java Fundamentals

Enumerations (Aufzählungstypen)

Enums

- Sind spezialisierte Klassen, die nur eine eingeschränkte Anzahl von Instanzen haben.
- Sie dienen dazu, Variablen mit eingeschränktem Wertebereich zu definieren.
- Anwendungsbeispiele: Monate, Wochentage, Innerer Zustand einer Klimaanlage (heizen, kühlen, aus)

Enums

```
public enum ACStatus {
   COOLING, HEATING, OFF
}
```

Die Enum „ACStatus“ hat 3 Instanzen, die über „COOLING“, „HEATING“ und „OFF“ referenziert werden.

Der Konstruktor einer Enum ist immer private. Es können keine zusätzlichen Instanzen erzeugt werden

```
public class Aircondition {
   private ACStatus status = ACStatus.OFF;
   private double measuredTemperature;
   private double targetTemp;
   private double heatingThresholdTemp;

   public void setMeasuredTemperature(double measuredTemperature){
      if (measuredTemperature > targetTemp + 2){
         status = ACStatus.COOLING;
      } else if (measuredTemperature < targetTemp -2){
         status = ACStatus.HEATING;
      } else {
         status = ACStatus.OFF;
      }
   }
  // getter and setter
}
```

Enums

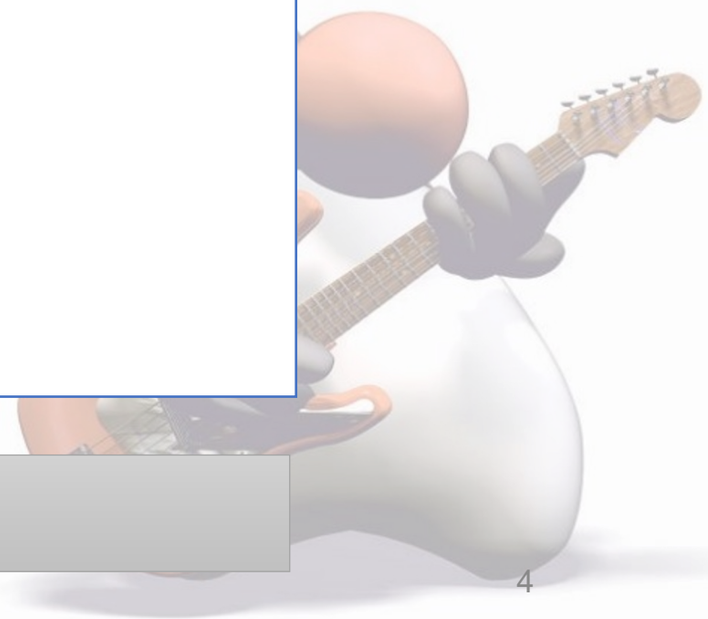
```
public enum ACStatus {
   COOLING, HEATING, OFF
}
```

Enums implementieren die toString Methode.

Enum Werte können sowohl mittels ==
als auch mittels .equals() verglichen werden.

Enum Variablen können den Wert null annehmen.

```
StringBuilder builder = new StringBuilder("Status Aircondition: ").append(status.toString()).append("\n");

if (status == null){
   builder.append("Unknown Status");
} else if (status == ACStatus.COOLING){
   builder.append("Aircondition is cooling");
} else if (status.equals(ACStatus.HEATING)){
   builder.append("Aircondition is heating");
} else {
   builder.append("Aircondition is OFF");
}
return builder.toString();
```



```
Status Aircondition: COOLING
Aicondition is cooling
```

Enums

Enums können in Switch Statements verwendet werden.

```
public enum ACStatus {
    COOLING, HEATING, OFF
}
```

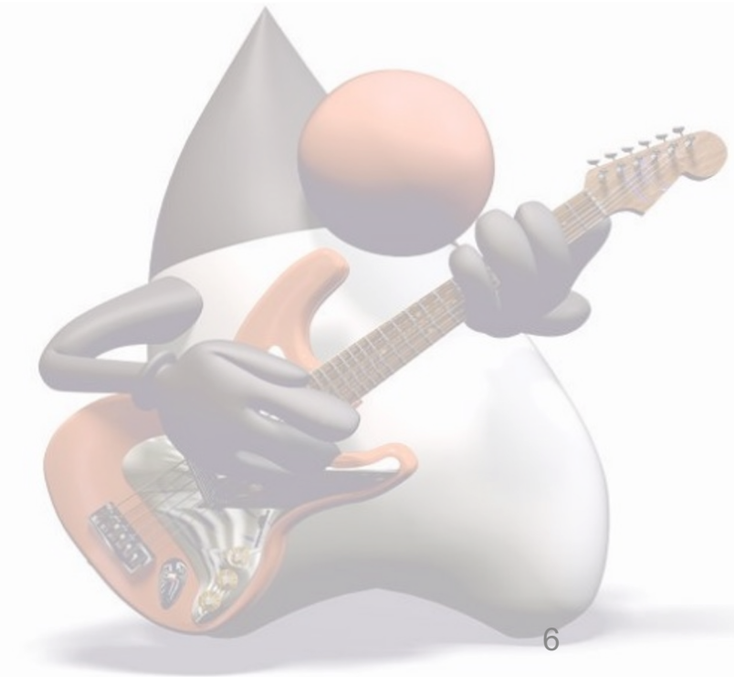
```
StringBuilder builder = new StringBuilder("Status Aircondition: ").append(status.toString());

switch (status) {
    case COOLING:
        builder.append("Aircondition is cooling");
        break;
    case HEATING:
        builder.append("Aircondition is heating");
        break;
    case OFF:
        builder.append("Aircondition is off.");
        break;
    default:
        builder.append("Aircondition state is invalid");
        break;
}

return builder.toString();
```

Enums

```
public enum ACStatus {
   COOLING, HEATING, OFF
}
```

Über values() kann man alle Instanzen der Enum ermitteln.

```
ACStatus[] allStatus = ACStatus.values();
for (ACStatus status: allStatus){
   System.out.println(status);
}
```



Enums

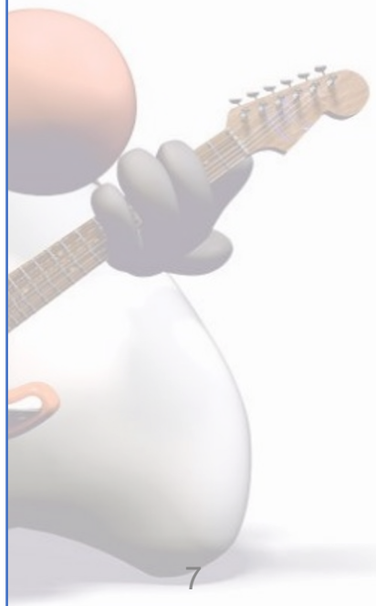
Da enums eigentlich Klassen sind,
können sie auch Attribute und Methoden haben.

Dadurch erweitert sich das Anwendungsgebiet der Enums ganz allgemein auf Klassen mit einer eingeschränkten Anzahl an Instanzen.

Der Konstruktor wird im Zuge der Definition der Enum Werte aufgerufen.

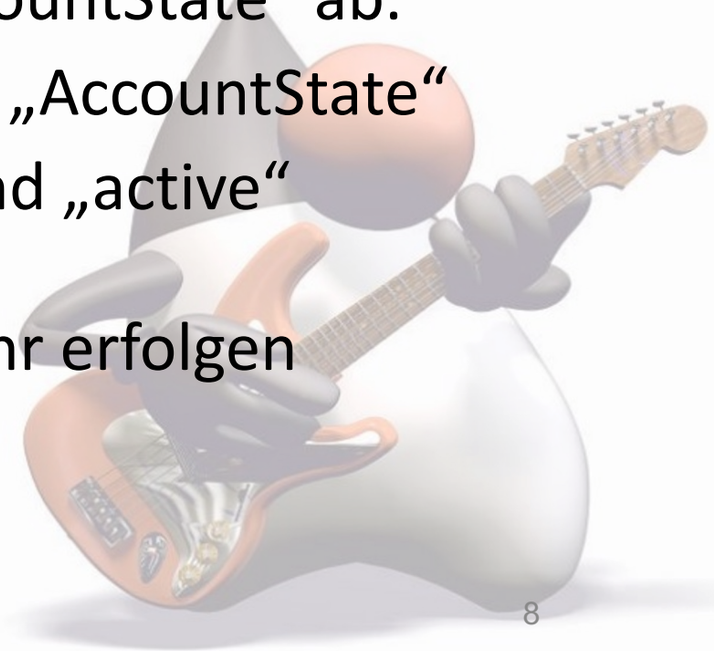
```
ACStatus status = aircondition.getStatus();
boolean isFanOn = status.isFanOn();
```

```
public enum ACStatus {
   COOLING(true, false, true),
   HEATING(true, true, false),
   OFF(false, false, false);

   private boolean fanOn;
   private boolean heaterOn;
   private boolean coolerOn;

   ACStatus(boolean fanOn, boolean heaterOn, boolean coolerOn) {
      this.fanOn = fanOn;
      this.heaterOn = heaterOn;
      this.coolerOn = coolerOn;
   }

   public boolean isFanOn() {
      return fanOn;
   }

   public boolean isHeaterOn() {
      return heaterOn;
   }

   public boolean isCoolerOn() {
      return coolerOn;
   }
}
```



Übungsaufgabe

- Ein Account soll die folgenden Zustände annehmen können:
  - created
  - active
  - locked
  - closed
- Bilde die Zustände eines Accounts über eine Enum „AccountState“ ab.
- Erweitere die Klasse Konto um das Feld „state“ vom Typ „AccountState“
- Verändere die Logik des Accounts so, dass nur im Zustand „active“ Abhebung durchgeführt werden können
- … und dass im Zustand „closed“ keine Einzahlungen mehr erfolgen können.



Java Fundamentals

Enumerations (Aufzählungstypen)