

OOP und File-I/O

Einleitung

Ein Motorradhändler will einen Teil seiner Motorräder auf einer Messe anbieten. Seine Transportkapazität ist auf vier Motorräder beschränkt, d.h. er kann nur eine Teilmenge der Produkte seines Ausstellungsraums transportieren.

Erfahrungsgemäß verkaufen sich auf derartigen Messen jene Motorräder am besten, die

- über die höchste Motorleistung verfügen oder
- das beste „Leistungsgewicht“ besitzen (Verhältnis von Leistung/Gewicht).

Der Händler wählt daher die vier Motorräder wie folgt aus:

- Die beiden stärksten Motorräder haben einen Fixplatz.
- Die restlichen beiden Plätze gehören jenen beiden Motorrädern mit dem größten „Leistungsgewicht“.

Aufgabenstellung

Einlesen der Datei `tuned_bikes.txt`

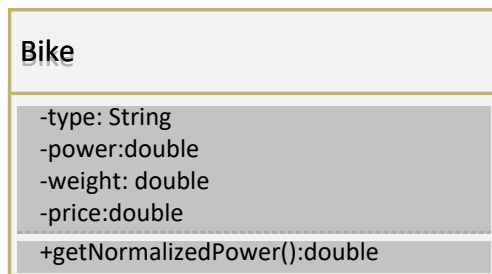
Ein Ausschnitt der Datei sieht wie folgt aus:

```
Bezeichnung:Leistung[PS]:Gewicht[kg]:Preis[€]  
KTM RC8 1190:175:184:22500  
HONDA FIREBLADE 1000RR:178:211:23000  
YAMAHA YZF-R1 1000:182:206:19999
```

Jede Zeile stellt dabei einen Datensatz bzw. Record dar. Die Datei ist zeilenweise einzulesen, jede Zeile ist mit : als Delimiter/Separator zu splitten. Zur Kontrolle sind die Daten mit folgendem Format auf der Konsole auszugeben:

```
Bezeichnung * Leistung[PS] * Gewicht[kg] * Preis[€]  
KTM RC8 1190 * 175 * 184 * 22500  
HONDA FIREBLADE 1000RR * 178 * 211 * 23000  
YAMAHA YZF-R1 1000 * 182 * 206 * 19999
```

Design und Implementierung der Klasse Bike



An die Klasse werden folgende Anforderungen gestellt:

- Attribute siehe Datei
- ein Konstruktor, der alle Werte entgegennimmt
- Getter-/Setter-Methoden für alle Attribute

- eine toString-Methode mit dem Format `Bike [type=SUZUKI HAYABUSA 1300]`
- eine getNormalizedPower-Methode zur Ermittlung des „Leistungsgewichtes“.

Erzeugung von Objekten

Alle Bikes obiger Datei sind in Objekten der entworfenen Klasse zu speichern, einmal in einem Array, anschließend in einer Liste.

Auswahl der vier Motorräder

In einem Konsolenprogramm sollen die vier Motorräder bestimmt und auf der Konsole im Format

```
Bike [type=SUZUKI HAYABUSA 1300]
```

```
Bike [type=DUCATI PANIGALE 1200]
```

```
...
```

```
...
```

ausgegeben werden.

Implementierungshinweise

Das folgende Codefragment illustriert das zeilenweise Einlesen einer Textdatei:

```
String path = "io/tuned_bikes.txt";
try {
    FileReader fr = new FileReader(path);
    BufferedReader br = new BufferedReader(fr);

    String line = null;
    String[] tokens = null;
    while((line = br.readLine()) != null){
        tokens = line.split(":");
        for(String token:tokens){
            System.out.print(token + " * ");
        }
        System.out.println();
    }
} catch (FileNotFoundException fnfe) {
    System.out.println("Please check the file location!");
} catch (IOException ioe) {
    System.out.println("Troubles reading data!");
}
```