

Hausaufgabe 7

Abgabe: 17.12.2023 (23:59 Uhr)

Abgabe:

Für dieses Hausaufgabenblatt sind **zwei Dateien abzugeben, *Clock.java* und *Car.java***. Zum Testen gibt es bereits den unten genannten Code als *ClockTest.java* bzw. *CarTest.java* für Sie in Moodle. Jede abgegebene Klasse muss für sich lauffähig sein. Nehmen Sie auch Quellcodebeispiele der SU als Basis, sofern hilfreich.

Aufgabe 0: Eine Uhr simulieren

In dieser Aufgabe sollen Sie eine Klasse *Clock* erstellen (welche eine simple Uhr repräsentiert). Eine Uhr zählt stundenweise und beginnt bei Stunde 0. Durch einen Aufruf der Methode *tick()* rückt die Uhr eine Stunde vor. Die Uhr gibt bei einem Aufruf von *print()* die aktuelle Tageszeit als Zeile auf der Konsole aus. Durch *switchMode()* kann die Uhr zwischen einer 12h Anzeige und einer 24h Anzeige hin und her wechseln. Im 12h Modus zeigt die Uhr Tageszeiten zwischen 1 und 12 mit dem Zusatz *am* (für vormittags) oder *pm* (für nachmittags), im 24h Modus zwischen 0 und 23 Uhr mit dem Zusatz *Uhr*¹. **Achtung: 0 Uhr nachts ist 12 am.** Durch einen Aufruf von *switchToSummertime()* schaltet die Uhr auf Sommerzeitanzeige um (1h mehr). Ein Aufruf von *switchToWintertime()* setzt dies wieder zurück. Ist die Uhr bereits in einem der Zustände (Sommer/Winter) bei Aufruf der Methoden, bleibt dieser Zustand einfach. Folgende Tabelle zeigt Code, der die Klasse *Clock* verwendet (links) und die davon erzeugte Konsolenausgabe (rechts). Den Code finden Sie in Moodle zum Download.

<pre>Clock basicClock = new Clock(); basicClock.print(); basicClock.tick(); basicClock.print(); for (int i = 1; i <= 12; i++) { basicClock.tick(); } basicClock.print(); basicClock.switchMode(); basicClock.print(); for (int i = 1; i <= 10; i++) { basicClock.tick(); } basicClock.print(); basicClock.switchMode(); basicClock.print(); basicClock.switchToSummertime(); basicClock.switchToSummertime(); basicClock.print(); basicClock.tick(); basicClock.print(); basicClock.switchToWintertime(); basicClock.print(); basicClock.switchMode(); basicClock.print();</pre>	<pre>Aktuelle Uhrzeit: 0 Uhr Aktuelle Uhrzeit: 1 Uhr Aktuelle Uhrzeit: 13 Uhr Aktuelle Uhrzeit: 1 pm Aktuelle Uhrzeit: 11 pm Aktuelle Uhrzeit: 23 Uhr Aktuelle Uhrzeit: 0 Uhr Aktuelle Uhrzeit: 1 Uhr Aktuelle Uhrzeit: 0 Uhr Aktuelle Uhrzeit: 12 am</pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

¹ siehe auch Tabelle unter <https://www.timeanddate.de/uhrzeit/am-pm-bedeutung>

Aufgabe 1: Eine Klasse zum Erzeugen eines Auto-Objektes

In dieser Aufgabe sollen Sie eine Klasse Car erstellen (diese repräsentiert ein Auto in einem fiktiven Videospiel). Ein Auto besitzt eine aktuelle Geschwindigkeit, eine Höchstgeschwindigkeit und eine Richtung („Norden“, „Süden“, „Westen“, „Osten“; es gibt vorerst nur diese vier Fahrtrichtungen, da das Straßennetz nur waagerecht oder senkrecht verlaufende Straßen kennt; wählen Sie eine passenden Datentyp für ein/mehrere Attribute, in welchen Sie sich diese Richtung speichern, *Tipp: String geht, ist aber umständlich*).

Ein neues Auto startet mit aktueller Geschwindigkeit 50, seine Höchstgeschwindigkeit wird im Konstruktor gesetzt. Bei Werten der Höchstgeschwindigkeit unter 50 soll 50 die Höchstgeschwindigkeit sein. Gibt man Gas, erhöht sich die Geschwindigkeit um 10, kann aber nicht größer werden als die Höchstgeschwindigkeit. Ein Auto kann nach links oder rechts seine Richtung ändern und man kann seinen Status als String abfragen (implementieren Sie dazu eine Methode `public String toString()` in Ihrer Klasse Car; diese wird auch automatisch von `System.out.println` verwendet, falls ein Car ausgegeben werden soll). Schreiben Sie die Klasse Car so, dass sie zum folgenden Beispielcode passt, welches Sie in einer main-Methode (einer anderen Klasse, z.B. CarTest, verwenden können). Den Beispielcode finden Sie in Moodle zum Download.

Dem Beispielcode können Sie die notwendigen Methoden entnehmen.

In den Kommentaren des Beispielcodes steht das gewünschte Ergebnis.

```
Car myCar = new Car(85); // Auto mit Höchstgeschwindigkeit 85
System.out.println(myCar); // cool, benutzt .toString() automatisch für die Ausgabe!
// ergibt in der Konsole: "Auto fährt mit Geschwindigkeit 50 Richtung Norden"
myCar.speedup(); // Gas geben; Geschwindigkeit jetzt 60
myCar.turnRight(); // Richtung jetzt: Osten
myCar.turnRight(); // Richtung jetzt: Süden
for (int i = 1; i <= 3; i++) {
    myCar.speedup();
    myCar.turnLeft();
}
System.out.println(myCar); // cool, benutzt .toString() automatisch für die Ausgabe!
// ergibt in der Konsole: "Auto fährt mit Geschwindigkeit 85 Richtung Westen"
```

(Knobelaufgabe auf der nächsten Seite)

Knobelaufgabe

Folgendes Programm sei der Beginn eines großen Programmierprojektes. Die Entwickler*innen hatten die Idee, Java zu verbessern und bestimmten Standard-Klassen neue Fähigkeiten zu verleihen. Dazu schreiben diese hier zunächst eine eigene Klasse *String*, die intern die bisherigen Strings benutzt und um neue Methoden erweitern soll. Der erste Versuch (s.u.) kompiliert auch, doch das Programm startet nicht!

1. Können Sie dem Projektteam helfen? Warum verweigert die JVM den Start des Programms?
2. Wie lässt sich das Problem hier lösen? Wie lässt sich so ein Problem künftig in ‚guten Softwareprojekten‘ komplett vermeiden?

```
public class StringNameReuse {  
    public static void main(String[] args) {  
        String hello = new String("Hello students, I am my own type of String!");  
        System.out.println(hello);  
    }  
}  
  
class String {  
    private java.lang.String myValue;  
  
    public String(java.lang.String text) {  
        this.myValue = text;  
    }  
    public java.lang.String toString() {  
        return myValue;  
    }  
}
```

Lösung: (markieren Sie die folgenden weißen Linien und kopieren diese; dann Einfügen in einem Text-Editor)

Lösung Ende.