







**wawi01** 

Vorführaufgabe 1

Name	Last update
 lib (/pg2/wawi01/tree/master/lib)	2 days ago
 src (/pg2/wawi01/tree/master/src)	2 days ago
 test (/pg2/wawi01/tree/master/test)	2 days ago
 .gitignore (/pg2/wawi01/blob/master/.gitignore)	2 days ago
 README.md (/pg2/wawi01/blob/master/README.md)	2 days ago

 **README.md** (/pg2/wawi01/blob/master/README.md)

# Entwicklung eines Kassensystems

In kleineren Shops und Restaurants haben Sie sicherlich schon einmal beobachtet, dass das Personal die Bestellungen und Abrechnungen mit Smartphones oder Tablets statt mit Registrierkassen oder PCs erledigen. Im Rahmen dieser und noch zwei darauf aufbauender Vorführaufgaben werden Sie ein stark vereinfachtest Kassensystem (auch "Point of Sale"-System, POS-System, genannt) in Java entwickeln.

## Vorführaufgabe 1

### Behandelte Themen

Definition von Klassen, Algorithmik allgemein, String-Methoden, `toString()`, (Wrapper-Klassen)

### Vorbereitungen

Sofern Sie sich mit dem CVS-Tool *git* auskennen, können Sie dieses Projekt mit folgendem Terminal-Befehl klonen:

```
git clone http://im-lamport.oth-regensburg.de:1080/pg2/wawi01.git
```

Hierfür müssen Sie das Tool *git* von folgender Website (<http://www.git-scm.com/download>) downloaden und installieren.

Ansonsten kopieren dieses Projekt auf ihren lokalen Rechner, so haben Sie bereits die entsprechenden JUnit-Testcases im Projekt und können lokal testen.

## Aufgabe 1: Klasse `Artikel` definieren

Definieren Sie eine Klasse `Artikel`, die folgende Eigenschaften berücksichtigt:

- Ein Artikel hat eine Nr
- Ein Artikel hat eine Bezeichnung
- Ein Artikel hat eine Kurzbezeichnung
- Ein Artikel hat einen ganzzahligen Lagerbestand
- Bei der Erzeugung von Objekten der Klasse `Artikel` müssen immer eine Nr und eine Bezeichnung übergeben werden; eine Anlage neuer Artikel ohne diese beiden Werte darf nicht möglich sein
- Eine Nr kann nur gelesen werden, später (d. h. nach Objekterzeugung bzw. Konstruktoraufruf) aber nicht mehr verändert werden
- Eine Bezeichnung kann jederzeit geändert werden
- Der Lagerbestand kann gelesen aber nicht direkt auf einen bestimmten Wert gesetzt werden. Vielmehr kann der Lagerbestand um eine gegebene Anzahl
  - erhöht („einlagern“) oder verringert („auslagern“) werden
  - Reicht der Lagerbestand beim Auslagern nicht vollständig aus, erfolgt zunächst keinerlei Änderung des Lagerbestands
- Die Kurzbezeichnung kann direkt nicht gesetzt werden
- Die Kurzbezeichnung wird automatisch aus der Bezeichnung generiert (s. Aufgabe 2). Ändert sich die Bezeichnung, ändert sich auch die Kurzbezeichnung

Wenden Sie die in der Vorlesung vorgestellten Namenskonventionen auf die Attribute und Methoden an. Achten Sie dabei darauf, wie die Attribute (erkennbar an der Formulierung „hat ein(e)“) im obigen Text bezeichnet wurden. Konstruktoren und Getter-/Setter-Methoden sind ebenfalls aus den o. g. Punkten abzuleiten.

Die `toString()`-Methode ist entsprechend des Beispiels im Abschnitt **Tests** unten zu gestalten.

## Aufgabe 2: Methode zur Erstellung von Kurzbezeichnungen

Die Klasse `Artikel` soll eine *statische* Methode

```
String erzeugeKurzbezeichnung(int nr, String bezeichnung)
```

erhalten, die aus einer übergebenen Nr und Artikelbezeichnung eine Kurzbezeichnung erstellt. Aufrufe dieser Methode sind fachlich passend in die Klasse zu integrieren (siehe hierzu Angaben in Aufgabe 1).

Eine Kurzbezeichnung eines Artikels besteht aus:

- bis zu 8 Konsonanten oder Ziffern aus der Artikelbezeichnung
- exakt 4 Ziffern, die sich aus der Artikel-Nr. ergeben
- einer 1-stelligen „Prüfziffer“

*Beispiel*

Attribut	Wert
----------	------

Attribut	Wert
Artikel-Nr.	123
Artikelbezeichnung:	Torx-Schrauben 6x35
Ergeben folgende Kurzbezeichnung:	TRXSCHRB01238

Folgende **Regeln** gelten für die Bildung der Kurzbezeichnung:

- Es werden die ersten 8 Konsonanten und/oder Ziffern von 0 - 9 der Artikelbezeichnung berücksichtigt
  - sind weniger vorhanden ist dieser Teil entsprechend kürzer
  - Sonderzeichen werden nicht berücksichtigt
  - Abweichende Alphabete mit Akzenten u. Ä. können ignoriert werden (diese Fälle werden nicht getestet)
  - Der Buchstabe „ß“ wird durch ein „s“ ersetzt
  - Die Kurzbezeichnung enthält nur Großbuchstaben
- Es werden die letzten 4 Stellen der Artikel-Nr. angefügt
  - Ist die Artikel-Nr. kürzer, wird mit der Ziffer „0“ auf 4 Stellen aufgefüllt
- Am Ende wird eine 1-stellige **Prüfziffer** angefügt

Die **Bildungsregeln für die Prüfziffer** lauten wie folgt:

- Jedem der 26 Buchstaben des Alphabets ist ein Wert zugeordnet: A = 1, B = 2, C = 3, ... Z = 26
- Die Werte aller Konsonanten sowie der einzelnen Ziffern (aus Bezeichnung und den letzten 4 Ziffern der Nr) werden zusammenaddiert und der ganzzahlige Rest einer Division der Summe durch 11 wird als Prüfziffer angehängt
  - Beträgt der ganzzahlige Rest 10, so ist ein „X“ anzuhängen
  - Beispiel:

```
SPX8X550008 = 19+16+24+8+24+5+5+0+0+0+8 = 109
109 % 11 = 10
10 → 'X'
```

Folgende Beispiele zeigen Kurzbezeichnungen sowie ihre Ausgangsgrößen:

Artikelbezeichnung	Artikel-Nr.	Prüfziffer	Kurzbezeichnung
Torx-Schrauben 6x35	123	8	TRXSCHRB01238
Spax 8x55	8	X	SPX8X550008X
Schloßschraube	62873408	7	SCHLSSCH34087

## Tests

Während der Entwicklung können Sie Ihre Ergebnisse zunächst mit Hilfe einer `main`-Methode testen. Der folgenden Code-Ausschnitt...

```
public static void main(String args[]){  
    // Objekte erzeugen und ausgeben  
    Artikel a = new Artikel(123, "Torx-Schrauben 6x35");  
    System.out.println(a);  
    Artikel b = new Artikel(8, "Spax 8x55");  
    b.einlagern(17);  
    System.out.println(b);  
  
    // Erzeugen der Kurzbezeichnung testen  
    System.out.println(erzeugeKurzbezeichnung(62873408, "Schloßschraube"));  
}
```

...erzeugt folgende Ausgabe auf der Konsole:

```
123, TRXSCHRB01238, Torx-Schrauben 6x35, 0 auf Lager  
8, SPX8X550008X, Spax 8x55, 17 auf Lager  
SCHLSSCH34087
```

## Abgabe

Die Abgabe und automatische Testierung erfolgt über den *Praktomat* (<http://praktomat3.oth-regensburg.de>), indem Sie dort Ihre entwickelte Klasse `Artikel.java` hochladen.

Sofern noch nicht geschehen, melden Sie sich hier (<http://praktomat3.oth-regensburg.de/accounts/register>) mit Ihrer OTH-Email-Adresse an.