

Thomas Fahle

## How To XXV

### Überprüfung von Nummerncodes mit `Algorithm::CheckDigits`

**Algorithm::CheckDigits** - Perl extension to generate and test check digits - von Mathias Weidner vereinfacht die Überprüfung von Nummerncodes, wie Umsatzsteuer-Identifikationsnummern, ISBN, Betriebsnummern, Blutbeutel-Eurocodes und zahlreichen mehr. Dabei bietet `Algorithm::CheckDigits` eine konsistente Schnittstelle zu allen Berechnungsmethoden an.

Mit `CheckDigits()` wird das gewünschte Berechnungsverfahren gewählt und man erhält das passende Objekt zurück. Die Methode `is_valid()` prüft den Nummerncode, `basenumber()` liefert die Basiszahl des Nummerncodes und `checkdigit()` liefert die Prüfwert für die Basiszahl zurück.

### Alle verfügbaren Berechnungsmethoden auflisten

Eine Übersicht aller verfügbaren Berechnungsmethoden liefert `method_list`:

```
#!/usr/bin/perl
use strict;
use warnings;

use Algorithm::CheckDigits;

my @ml =
    Algorithm::CheckDigits->method_list();

foreach my $method ( @ml ) {
    print "$method\n";
}
```

Das Programm liefert die folgende Ausgabe (deutlich gekürzt):

```
2aus5
ahv_ch
amex
bahncard
betriebsnummer
blutbeutel
...
ustid_at
ustid_be
ustid_de
...
verhoeff
visa
wagonnr_br
```

### Beispiel: Umsatzsteuer-Identifikationsnummer

Dieses Beispiel testet eine deutsche Umsatzsteuer-Identifikationsnummer (`ustid_de`).

```
#!/usr/bin/perl
use strict;
use warnings;

use Algorithm::CheckDigits;

my $ustid_de = CheckDigits('ustid_de');

#Umsatzsteuernummer: DE175903868
my $ustid = '175903868';

if ( $ustid_de->is_valid($ustid) ) {
    print "UmsatzsteuerID: $ustid ist okay.\n";
    my $cd = $ustid_de->checkdigit($ustid);
    my $bn = $ustid_de->basenumber($ustid);
    print "Checkdigit: $cd\n";
    print "Basiszahl: $bn\n";
}
else {
    print
        "UmsatzsteuerID: $ustid ist NICHT okay.\n";
}
```



Das Programm liefert folgende Ausgabe:

```
UmsatzsteuerID: 175903868 ist okay.  
Checkdigit: 8  
Basiszahl: 17590386
```

Umsatzsteuer-Identifikationsnummern aus Österreich lassen sich mit der Methode `ustid_at` checken. Ebenso lassen sich Umsatzsteuer-Identifikationsnummern aus den Niederlanden mittels `ustid_nl` oder aus Belgien via `ustid_be` testen.

## Beispiel: Internationale Standardbuchnummer (ISBN)

Das folgende Beispiel testet eine 10-stellige ISBN (`isbn`). 13-stellige ISBN lassen sich mit `isbn13` prüfen.

```
#!/usr/bin/perl  
use strict;  
use warnings;  
  
use Algorithm::CheckDigits;  
  
my $isbn10_cd = CheckDigits('isbn');  
  
# Einführung in Perl  
my $isbn = '386899145X';  
  
if ( $isbn10_cd->is_valid($isbn) ) {  
    print "ISBN-10: $isbn ist okay.\n";  
    my $cd = $isbn10_cd->checkdigit($isbn);  
    my $bn = $isbn10_cd->basenumber($isbn);  
    print "Checkdigit: $cd\n";  
    print "Basiszahl: $bn\n";  
}  
else {  
    print "ISBN-10: $isbn ist NICHT okay.\n";  
}
```

Das Programm liefert folgende Ausgabe:

```
ISBN-10: 386899145X ist okay.  
Checkdigit: X  
Basiszahl: 386899145
```

## Weitere Nummerncodes

Algorithm::CheckDigits stellt aktuell **127** Testverfahren nach dem gezeigtem Schema bereit. Ich empfehle einen Blick in die Dokumentation: Der gesuchte Nummerncode ist mit hoher Wahrscheinlichkeit bereits implementiert.