



Zentrum für
Informationsverarbeitung
und Informationstechnik

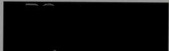


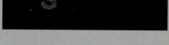
Steueridentifikations- nummer (IdNr) nach § 139b AO

Informationen zur Berechnung gültiger Prüfziffern


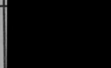
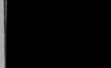
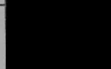
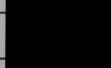
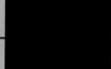
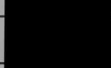
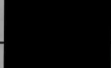
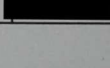


KONZEPT

Informationen zum vorliegenden Dokument

Dokumententitel	Steueridentifikationsnummer (IdNr) nach § 139b AO. Informationen zur Berechnung gültiger Prüfziffern		
Version	2.0		
erstellt am	14.08.2014	von	
zuletzt bearbeitet am	16.08.2014	von	
QS-geprüft		von	
freigegeben am	19.08.2014	von	
vorgelegt am	19.08.2014		

Änderungshistorie

<u>Version</u>	<u>Datum</u>	<u>Änderung</u>	<u>bearbeitet von</u>
2.0-0	14.08.2014	<i>Übernahme:</i> Version vom 20.03.2008	
2.0-1	15.08.2014	<i>Erweiterung:</i> Zusätzliche Beschreibung der dreistelligen Ziffernwiederholung; Anpassung der definierten IdNr-Eigenschaften.	
2.0-2	16.08.2014	<i>Überarbeitung:</i> Verbesserte Darstellung der JAVA-Methode.	
	18.08.2014	Review	
2.0-3	18.08.2014	Einarbeitung Review	
	18.08.2014	Review	
2.0-4	18.08.2014	Einarbeitung Review	
	19.08.2014	Review	
2.0-5	19.08.2014	Einarbeitung Review	

Inhaltsverzeichnis

1. EIGENSCHAFTEN DER IDENTIFIKATIONSNUMMER	4
2. BEISPIELE GÜLTIGER IDNR	4
3. BERECHNUNG DER PRÜFZIFFER	5
3.1 Allgemeiner Algorithmus zur Berechnung der Prüfziffer	5
3.2 JAVA-Methode zur Berechnung der Prüfziffer	6

1. Eigenschaften der Identifikationsnummer

Für gültige Steueridentifikationsnummern (IdNr) nach § 139b AO gelten folgende festgelegte Eigenschaften:

Nr.	Eigenschaft
1	Die IdNr besteht aus einer elfstelligen Ziffernfolge.
2	An erster Position steht <u>niemals</u> eine Null.
3	An der elften Position steht eine Prüfziffer.
4	An den Positionen 2 bis 10 stehen Ziffern aus der Menge {0,1,2,3,4,5,6,7,8,9}.
5	Unter den Positionen 1 bis 10 befindet sich <u>immer</u> eine Zifferwiederholung. <ul style="list-style-type: none"> 5.1 <u>Entweder</u> genau zwei Ziffern sind gleich <u>oder</u> genau drei Ziffern sind gleich. 5.2 Es gibt niemals mehr als drei gleiche Ziffern. Dies gilt ausschließlich für die ersten 10 Ziffern, nicht aber für die Prüfziffer. 5.3 Existieren <u>zwei</u> gleiche Ziffern an den ersten zehn Positionen, können diese an beliebigen Stellen (unter den Positionen 1 bis 10 verteilt) stehen. Die restlichen acht Ziffern treten jeweils einmalig auf. 5.4 Existieren <u>drei</u> gleiche Ziffern an den ersten zehn Positionen, dürfen diese gleichen Ziffern <u>niemals</u> an direkt aufeinander folgenden Stellen stehen. 5.5 Existieren <u>drei</u> gleiche Ziffern an den ersten zehn Positionen, können diese an beliebigen Stellen (unter den Positionen 1 bis 10 verteilt) stehen, solange sie nicht an direkt aufeinander folgenden Stellen stehen. Die restlichen sieben Ziffern treten jeweils einmalig auf.

2. Beispiele gültiger Identifikationsnummern

Entsprechend der oben definierten Eigenschaften sind folgende beispielhafte Zifferkombinationen gültig für das Format einer IdNr:

IdNr inkl. Prüfziffer	Prüf- ziffer	Doppelte Ziffer	Dreifache Ziffer
86095742719	9	7	-
47036892816	6	8	-
65929970489	9	-	9
57549285017	7	-	5

3. Berechnung der Prüfziffer

An elfter Ziffer-Position der IdNr steht eine Prüfziffer. Sie soll mögliche Fehler bei angegebenen IdNr maschinell überprüfbar machen. Solche Fehler können durch Versehen bei manuellen Eingaben, falsch verarbeiteten OCR-Scans oder wegen mangelhafter Datenübertragung entstehen. Hierbei kommt es häufig zu unrichtigen Verdopplungen von Ziffern, fehlenden Ziffern oder sogenannten „Zahlendrehern“¹.

Bei der IdNr-Prüfziffer wird eine kombinierte Berechnung angewendet, die auf einer modifizierten, gemeinsamen Verwendung des „Modulus 10“ und „Modulus 11“ aufbaut. Mit dem hier beschriebenen Verfahren soll eine möglichst hohe Quote der Fehlererkennung erreicht werden.

3.1 Allgemeiner Algorithmus zur Berechnung der Prüfziffer

- c_{pos} steht für eine der Ziffern an den Positionen 1 (c_1) bis 10 (c_{10}).
- pz ist die Prüfziffer.
- pos, m11, m10 sind Hilfsvariablen.

Die Prüfziffer berechnet sich nach folgendem Algorithmus:

```
begin
  m11 := 10
  m10 := 0
  for pos = 1 to 10 step 1
    m10 := ( $c_{pos}$  + m11) mod 10
    if m10 = 0
      then m10 := 10
    end-if
    m11 := (2 * m10) mod 11
  end-for
  pz := 11 - m11
  if pz = 10
    then pz := 0
  end-if
end.
```

¹ vgl. Zelt, Kurt (2002): Prüfzifferverfahren. Diplomarbeit an der Verwaltungs- und Wirtschafts-Akademie Rhein-Neckar e.V. Mannheim. Seite 7f. Abgerufen am 16.08.2014 unter <http://www.kurt-zelt.de/files/PZV.pdf>

3.2 JAVA-Methode zur Berechnung der Prüfziffer

Die angegebene JAVA Methode berechnet die Prüfziffer für eine vorgegebene Steueridentifikationsnummer (IdNr). Die Prüfziffer-Ermittlung basiert auf einer Berechnung des Restwerts ganzzahliger Divisionen. Hierbei wird jede Ziffer der ersten zehn Positionen berücksichtigt. Es werden zwei Modulo-Operationen in Kombination angewendet, die jeweils in Prüfziffer-Berechnungen verbreitet eingesetzt sind.

Für jede Ziffer auf den Positionen 1 bis 10 werden Restwerte aus ganzzahligen Divisionen durch 10 und durch 11 berechnet. Diese Berechnungen erfolgen in aufeinander aufbauenden Schritten. Das Ergebnis jedes Berechnungsschritts fließt in den Folgeschritt ein, so dass als Ergebnis die berechnete Prüfziffer an Position 11 von allen vorigen Ziffern der Positionen 1 bis 10 direkt abhängt. Damit ist ein direkter Bezug zur Position und Wertigkeit jeder Ziffer gewährleistet. Fehlerhafte Eingaben können so mit begrenztem Rechenaufwand und zutreffend erkannt werden.

```
/**
 * This method calculates the check digit for a German tax identification
 * number (IdNr).
 * Based on finding the division remainder (modulo calculation) the check
 * digit depends on each preceding numeric digit on position 1 to 10.
 * The chosen way of calculation combines and applies two commonly used modulo
 * operations. For each digit on positions 1 to 10 the remainders of both
 * divisions by eleven and ten are used in consecutive arithmetic steps.
 * The calculation result must correspond with the numeric digits on position 11.
 * This approach considers each digit in a specific way for being able
 * to detect falsely edited IdNr in a reliable and efficient way.
 *
 * @param _idnrString -- a string consisting of ten numeric digits
 * @return -- calculated check digit returned as an integer value
 */
public int calculateCheckDigit(String _idnrString) {

    final int ten = 10;
    final int eleven = 11;

    char[] chars = _idnrString.toCharArray();

    int remainder_mod_ten = 0;
    int remainder_mod_eleven = ten;
    int digit = 0;
    final int length = _idnrString.length();
    for (int counter=0; counter<length; counter++) {
        digit = Character.getNumericValue(chars[counter]);
        remainder_mod_ten = (digit + remainder_mod_eleven) % ten;
        if (remainder_mod_ten == 0)
            remainder_mod_ten = ten;
        remainder_mod_eleven = (2*remainder_mod_ten) % eleven;
    } // for

    digit = eleven - remainder_mod_eleven;
    if (digit == 10)
        digit = 0;

    return digit;

} // calculateCheckDigit
```

Einige Beispiele valider IdNr mit Prüfziffer finden sich unter Abschnitt 2.