

SEW I - Übung: Sozialversicherungsnummer-Prüfung

Ziel

Implementierung eines simplen Programms zur Überprüfung von österreichischen Sozialversicherungsnummern.

Lernziele

- Arbeiten mit Methoden.
- Arbeiten mit *Strings*, *chars* und der *ASCII*-Tabelle.

Abgaberichtlinien

- Ihre implementierte Lösung als `.java`-Datei. **Vergessen Sie dabei nicht auf Kommentare und Kommentarkopf!**
- Achten Sie auf **saubere Variablenbenennung** und **Nutzung von Konstanten** wenn sinnvoll!

Aufgabe

Hintergrund

Die österreichische Sozialversicherungsnummer besteht aus 10 Ziffern: Eine dreistellige laufende Nummer, eine Prüfziffer und das sechstellige Geburtsdatum (*TTMMJJ*). So wäre beispielsweise *1234010180* eine gültige Sozialversicherungsnummer für eine Person die am 1. Jänner 1980 geboren ist.

Die vierte Stelle, im obigen Beispiel die *4*, wird dabei aus den anderen Stellen berechnet. Dies geschieht folgendermaßen:

1. Jede Stelle der Sozialversicherungsnummer wird mit einem bestimmten Gewicht multipliziert. Für die Laufnummer sind die Gewichte *3*, *7* und *9*, für das Geburtsdatum *5*, *8*, *4*, *2*, *1* und *6*.
2. Anschließend werden die Produkte summiert.
3. Die Prüfziffer errechnet sich aus dem Divisionsrest (*Modulo*) dieser Summe geteilt durch *11*.
4. Ergibt die Prüfziffer *10*, so wird die Sozialversicherungsnummer nicht vergeben.

Folgendes Beispiel für eine am 2. Juli 1997 geborene Person soll Ihnen die Berechnung verdeutlichen:

Nummer	7	4	1	P	0	2	0	7	9	7
Gewichtung	3	7	9		5	8	4	2	1	6
Produkte	21	28	9		0	16	0	14	9	42
Summe der Produkte	139									

Implementierung

Zu erstellen ist ein Programm, das in der Lage ist Sozialversicherungsnummern auf ihre Gültigkeit zu überprüfen sowie gültige Prüfziffern zu generieren. **Implementieren Sie dabei zuerst die folgenden Methoden und testen Sie diese mit Testdaten durch Aufrufe aus Ihrer main-Methode.** Sobald Sie die Richtigkeit der einzelnen Methoden sichergestellt haben, können Sie diese verwenden um das finale Programm (siehe Beispiel unten) inklusive Benutzereingaben zu realisieren.

- `containsOnlyDigits` akzeptiert eine Zeichenkette als Parameter und gibt zurück (`boolean`), ob diese ausschließlich aus Ziffern besteht.
- `calculateCheckDigit` akzeptiert zwei Zeichenketten - die Laufnummer und das Geburtsdatum - und berechnet auf Basis der oben beschriebenen Vorgangsweise die Prüfziffer, die anschließend zurückgegeben wird.
 - Eine elegante Möglichkeit die Gewichte mit den entsprechenden Ziffern zu multiplizieren ist die Darstellung als `int-Array`. Dieses können Sie auch als statische Konstante speichern.
- `isSvnrValid` akzeptiert eine Sozialversicherungsnummer als Zeichenkette und überprüft, ob diese gültig ist. Verwenden Sie dafür Ihre Methode `calculateCheckDigit`, an die Sie die entsprechenden Teile der Sozialversicherungsnummer übergeben. Sie können die `String`-Methode `substring` verwenden um die Zeichenkette aufzuteilen.
 - Vergessen Sie natürlich nicht zu überprüfen, ob die Länge der Zeichenkette korrekt ist und diese ausschließlich aus Ziffern besteht.

Beispiele

Beispiel 1

```
=====
= Austrian Social Security Number Checker =
=====
Please select mode ('c' for check, 'g' for generate): G
Please enter your 3-digit serial number: 567
Please enter your birth date in DDMMYY-format: 230888
Your check digit is 6.
```

Beispiel 2

```
=====
= Austrian Social Security Number Checker =
=====
Please select mode ('c' for check, 'g' for generate): c
```

Please enter your 10-digit social security number: 7417020797
Your social security number is valid.

Beispiel 3

```
=====
= Austrian Social Security Number Checker =
=====
Please select mode ('c' for check, 'g' for generate): C
Please enter your 10-digit social security number: 1235010180
Your social security number is invalid.
```

Beispiel 4

```
=====
= Austrian Social Security Number Checker =
=====
Please select mode ('c' for check, 'g' for generate): c
Please enter your 10-digit social security number: 12345678901
Your social security number is invalid.
```