# Euler-001

***Wenn wir die natürlichen Zahlen kleiner 10 betrachten, die Vielfache von 3 oder 5 sind, erhalten wir 3, 5, 6 und 9; ihre Summe ist 23. Gesucht sind alle Vielfachen von 3 oder 5 kleiner 1000.***

Die nächstliegende Lösung besteht darin, mit einer Zählschleife alle Zahlen von 1 bis n = 999 zu durchlaufen und die Zahlen aufzusummieren, die Vielfache von 3 oder 5 sind.

Wenn n aber größer angenommen wird, kann das Erscheinen der Lösung zu lange dauern oder übermäßig viel Speicherplatz beanspruchen. Es empfiehlt sich daher, sich des jungen Gauß zu entsinnen. Schon in seiner Jugend zeigte sich bei Carl Friedrich Gauß (1777–1855) die mathematische Begabung. In der dritten Volksschulklasse, also im Alter von acht Jahren, demonstrierte er seine Fähigkeiten auf eindrucksvolle Weise. Der Lehrer Johann Georg Büttner hatte der Klasse die Aufgabe gestellt, die Zahlen von 1 bis 100 zusammenzuzählen. Gauß tat dies auf schnelle und elegante Weise, indem er 50 Paare mit der Summe 101 bildete (1 + 100, 2 + 99, …, 50 + 51) und 50 · 101 = 5050 als Ergebnis ablieferte.


**Carl Friedrich Gauß auf einem 10-DM-Schein**
**(mit Göttingen und der Gaußschen Normalverteilung im Hintergrund).**

Die allgemeine Gauß-Formel lautet

(1) $s(n) = 1 + 2 + \ldots + n = n \cdot (n + 1) / 2.$

Damit ist eine effiziente Möglichkeit gegeben, Summen zu berechnen. Handelt es sich um die Menge V(k, n) der Vielfachen von k, die kleiner n sind:

$$s(k, n) = \sum V(k, n) = k + 2k + 3k + \ldots,$$

muss die Formel etwas verallgemeinert werden. Im Fall von k = 3, n = 99 haben wir

$$s(3, 99) = 3 + 6 + \ldots + 99 = 3 \cdot (1 + 2 + \ldots + 33) = 3 \cdot s(div(99, 3)),$$

wobei *div(n, k)* die ganzzahlige Division von n durch k bezeichnet (in Smalltalk n // k).
Für k = 5 ergibt sich

$$s(5, 99) = 5 + 10 + \ldots + 95 = 5 \cdot (1 + 2 + \ldots + 19) = 5 \cdot s(div(99, 5)).$$

Allgemein:

(2) $s(k, n) = k \cdot s(div(n, k))$.

Nun können wir aber nicht einfach s(3, 99) + s(5, 99) bilden, denn es ist

$$\sum[V(3, n) \cup V(5, n)] \neq \sum V(3, n) + \sum V(5, n).$$

Das „oder“ der Aufgabenstellung bezeichnet die Vereinigungsmenge der Vielfachenmengen von 3 und 5. Das *Ein- und Ausschalt-Prinzip* (engl.: inclusion-exclusion principle) sagt:

$$\sum[V(3, n) \cup V(5, n)] = \sum V(3, n) + \sum V(5, n) - \sum[V(3, n) \cap V(5, n)].$$

In unserem Fall ist aber $\sum[V(3, n) \cap V(5, n)] = \sum V(15, n)$, wir haben also

(3) $s(3, n) + s(5, n) - s(15, n)$

zu berechnen (die fettgedruckten Vielfachen sind nur einmal zu berücksichtigen):

[3, 5, 6, 9, 10, 12, **15**, 18, 20, 21, 24, 25, 27, **30**, 33, 35, 36, 39, 40, 42, **45**, 48, 50, 51, 54, 55, 57, **60**, 63, 65, 66, 69, 70, 72, **75**, 78, 80, 81, 84, 85, 87, **90**, 93, 95, 96, 99]

Nun geht es um die Implementation der Funktionen *s(k, n)*. Die erste Möglichkeit besteht darin, zuerst Formel (1) und sodann Formel (2), die auf (1) aufbaut, zu realisieren.

Die Gauß-Formel (1) lautet in Squeak:

```
summeVon1BisN
  ^ self * (self + 1) // 2
```

Der Aufruf im Workspace:

```
100 summeVon1BisN --> 5050
```

Formel (2) implementiert lautet:

```
summeVon1BisNmitK: schrittweite
  ^ schrittweite * (self // schrittweite) summeVon1BisN
```

Das Programm zur Lösung der Aufgabe, also Formel (3) entsprechend, lautet dann:

```
n := 999.
(n summeVon1BisNmitK: 3)
+ (n summeVon1BisNmitK: 5)
- (n summeVon1BisNmitK: 15)
```

Die zweite Möglichkeit wäre eine direkte Implementation einer arithmetischen Reihe:

*Die Summe von N Zahlen gleichen Abstands mit dem ersten Glied a und dem letzten Glied z ist N-mal ihr Durchschnitt; in Formeln: $N \cdot (a + z) / 2$.*

In unserem Fall ist der Abstand und das erste Glied jeweils k, die Anzahl N ist div(n, k), und das letzte Glied ist $k \cdot div(n, k)$. Das ergibt, in Squeak implementiert:

```
arithmetReiheBis: grenze
```

```
  | erstes abstand anzahl letztes |
  abstand := erstes := self.
  anzahl := grenze // abstand.
  letztes := abstand * anzahl.
  ^ anzahl * (erstes + letztes) // 2
```

Der Aufruf im Workspace liefert

```
1 arithmetReiheBis: 100 --> 5050
```

Man beachte, dass jetzt die Reihenfolge der Argumente vertauscht ist. Das Programm zur Lösung von **E-001** lautet nunmehr:

```
n := 999.
(3 arithmetReiheBis: n)
+ (5 arithmetReiheBis: n)
- (15 arithmetReiheBis: n)
```

Das Euler-Problem E-001 wird erst eigentlich dadurch interessant, als es zu weiterführenden Überlegungen herausfordert.

**Aufgabe:** Man bestimme Anzahl und Summe der natürlichen Zahlen unter $10^n$, die weder durch 2, noch durch 3, noch durch 7 teilbar sind. (Die Grenze n ist dann geeignet hoch anzusetzen.)

Ferner sind Überlegungen anzustellen, wenn die vorgegebenen Zahlen k nicht teilerfremd zueinander sind.