Universität Leipzig **WS 14/15**
Institut für Informatik **Dr. Monika Meiler**

Modellierung und Programmierung 1
Übungsserie 2

Lösungsvorschläge

1. **Ausdrücke**

   siehe **Aresco.java**.

2. **Elementardatentypen**

   a) Induktionsanfang $n = 0$:

   $$s(0) = \sum_{k=0}^{0} k = \frac{0+1}{2} = 0$$

   Induktionsvoraussetzung:

   $$s(n) = \sum_{k=0}^{n} k = \frac{n(n+1)}{2}$$

   Induktionsbeweis:

   $$s(n+1) = \sum_{k=0}^{n+1} k = \sum_{k=0}^{n} k + (n+1)$$

   laut Induktionsvoraussetzung

   $$s(n+1) = \frac{n(n+1)}{2} + (n+1) = \frac{n(n+1) + 2(n+1)}{2} = \frac{(n+1)(n+2)}{2}$$

   b) siehe **Gauss.java**.

   c) Betrachte zunächst $s(n)$

   $$s(n) = \frac{n(n+1)}{2} = \frac{n^2+n}{2}$$

   $$\frac{n^2}{2} + \frac{n}{2} - s(n) = 0$$

   $$n^2 + n - 2 \cdot s(n) = 0$$

   Lösung der quadratischen Gleichung unter der Bedingung, dass $n > 0$

   $$n = -\frac{1}{2} + \sqrt{\frac{1}{4} + 2 \cdot s(n)}$$

   Wegen $s(n) \leq \texttt{Long.MAX\_VALUE} = 2^{63} - 1$ gilt

   $$\begin{aligned} n &\leq -\frac{1}{2} + \sqrt{\frac{1}{4} + 2 \cdot (2^{63} - 1)} \\ &= -\frac{1}{2} + \sqrt{\frac{1}{4} + (2^{64} - 2)} \\ &= -\frac{1}{2} + \sqrt{2^{64} - \frac{7}{4}} \end{aligned}$$

Aufgerundet auf die nächste natürliche Zahl ($n \in \mathbb{N}$) ergibt sich

$$n < -\frac{1}{2} + \sqrt{2^{64}} = -\frac{1}{2} + 2^{32} < 2^{32} - 1$$

Das Programm **Gauss.java** liefert für $n = 2^{32} - 1$ noch einen korrekten Wert, für $n = 2^{32}$ findet bereits ein Überlauf statt. Folglich ist das eine sehr gute Abschätzung!

d) Eine Deklaraton von s als `double` kann keine Verbesserung liefern, denn bei $n = 2^{32} - 1$ folgt

$$s(n) = \frac{(2^{32} - 1) \cdot 2^{32}}{2} = 2^{63} - 2^{31}.$$

Die Kapazität von 53 bit-Mantissen wird somit überschritten.
Bei `long`-Größen sind 19 Stellen sicher darstellbar, bei `double`-Größen sind es nur 15.

3. **Schleifenanweisung/Auswahlanweisungen**

   a) siehe **Collatz.java**.

   b) siehe **Collatz1.out**, **Collatz2.out** und **Collatz3.out**.
   Die letzten drei Folgenglieder für alle $n \in \mathbb{N}$ sind vermutlich immer 4, 2 und 1.

4. **Schleifenanweisung/Auswahlanweisungen**

   a) siehe **Goldbach.java**.

   b) siehe **Goldbach.out**.