Arithmetik – Kontrollstrukturen – Eingabe von Konsole lesen

A Summen berechnen

Schreiben Sie ein Programm, das die nachfolgenden Formeln berechnet.

1. $\sum_{i=1}^{n} \frac{1}{i^2+1}$

2. $\sum_{i=0}^{n} \frac{(x+i)}{(x^3+i)}$

3. $\sum_{i=1}^{n} \frac{\sqrt[i]{x}}{\sum_{j=1}^{i} \frac{(x-j)}{(x+j)}}$; schreiben Sie zuerst eine Funktion für den Summenausdruck im Nenner und setzen Sie diese Funktion in die Berechnung der äußeren Summe ein.

- Schreiben Sie für jede Formel eine eigene Methode.
- Beachten Sie die Effekte der ganzzahligen Division!
- Wenn der Wert für n den geforderten Wertebereich verletzt oder keine ganz Zahl ist, dann soll der Fehler ausgegeben, die Berechnungen sofort abgebrochen und als Ergebnis **-99** zurückgeben werden.
  Hinweis: ***x.integer?()*** liefert ***true***, wenn x eine ganze Zahl ist.
- Wenn x in 3. eine ganze Zahl ist, dann soll eine Fehler ausgegeben werden, die Berechnung sofort abgebrochen und **-99** zurückgegeben werden.
- Schreiben Sie für jede der 3 Methoden eine Schleife, die die Methode für 1000 unterschiedliche n aufruft und das Ergebnis ausgibt.
- Schreiben Sie für die Methoden aus 2. eine Schleife, die die Methode für 1000 unterschiedliche n mit ganzzahligem x aufruft und das Ergebnis ausgibt. Sind die Ergebnisse korrekt?
- Zeigen Sie für alle Methoden, dass die Wertebereichsprüfungen korrekt arbeiten.

B Satellitenzeit

Ein Satellit funkt Zeitspannen als „Anzahl Sekunden“ zur Erde. Schreiben Sie eine Methode ***satellite_time***, die die Sekunden in der Form **d h m s** ausgibt. Dabei sind

d = Anzahl Tage
h = Anzahl Stunden im Bereich 0..23
m = Anzahl Minuten im Bereich 0..59
s = Anzahl Sekunden im Bereich 0 bis 59

Die Sekunden werden der Methode als Parameter n übergeben. Wenn n keine positive ganze Zahl ist, dann gibt die Methode geeignete Fehlermeldungen aus und wird sofort beendet. Die Anzahl Sekunden soll von der Konsole eingelesen und in eine ganze Zahl umgewandelt werden. Danach soll die Methode ***satellite_time*** aufgerufen werden.

Beispiel:

```
satellite_time(17000)

0 4 43 20

satellite_time(100000)

1 3 46 40
```

Hilfestellung: Einlesen von Zahlen von der Konsole

zahl = gets().chomp().to_f()

gets() wartet auf die Benutzereingabe
chomp() entfernt den Zeilenumbruch
to_f() wandelt eine Zeichenkette in eine Gleitkommazahl, wenn der erste Teil der Zeichenkette ein gültiger Ausdruck für eine Zahl ist.

zahl = gets().chomp().to_i()

to_i() wandelt eine Zeichenkette in eine ganze Zahl, wenn der erste Teil der Zeichenkette ein gültiger Ausdruck für eine Zahl ist.

Wenn Sie beim Einlesen der Eingabe eine ganze Zahl erwarten und gleichzeitig damit rechnen müssen, dass eine Gleitkommazahl eingegeben wird, müssen Sie die Eingabe mit ***to_f()*** und ***to_i()*** konvertieren und anschließend die Ergebniszahlen mit ***==*** vergleichen. Wenn der Vergleich ***== true*** ergibt, dann wurde eine ganze Zahl eingegeben. Beim Einlesen darf aus einer Gleitkommazahl keine ganze Zahl werden!

to_f() und ***to_i()*** sind keine „echten“ Konvertierungen von Eingaben in Zahlen, da Sie z.B. die leere Zeichenkette in die Zahl 0 verwandeln. Korrekte Konvertierungen mit ***Integer*** oder ***Float*** können wir hier nicht verwenden, da wir noch keine Fehlerbehandlung in Ruby kennen!

C Primfaktoren

Schreiben Sie eine Methode ***prime_factors***, die zu einem übergebenen n ≥ 2 die Primfaktorenzerlegung berechnet. Beginnend mit 2 werden alle möglichen Teiler von n ausprobiert. Ein Faktor kann dabei mehrmals enthalten sein. Wenn ein echter Teiler gefunden wurde (verwenden Sie die Modulus Methode), wird dieser ausgegeben, n um den Teiler reduziert und die Zerlegung fortgeführt. Die Zerlegung endet, wenn n den Wert 1 erreicht. Wenn n < 2 oder n keine ganze Zahl, dann soll die Berechnung mit einem passenden Fehlerhinweis abgebrochen werden.

Zeigen Sie, dass Ihre Lösung für die Beispiele korrekte Ergebnisse liefert.

Beispiele:

prime_factors(1024)

```
2 2 2 2 2 2 2 2 2 2
```

prime_factors(1764)

```
2 2 3 3 7 7
```

prime_factors(1)

```
Fehler: 1 muss >= 2 sein
```

prime_factors(23.8)

```
Fehler: 23.8 ist keine ganze Zahl
```

D Perfekte Zahlen

Eine perfekte Zahl ist eine Zahl, die die Summe aller ihrer ganzzahligen Teiler (inklusive der 1) ist. Schreiben Sie eine Methode ***perfekt?(n)***, die zu einer ganzen positiven Zahl bestimmt, ob diese Zahl eine perfekte Zahl ist. Die Methode liefert ***true***, wenn n eine perfekte Zahl ist, sonst ***false***. Nutzen Sie diese Methode in einer Schleife, um drei weitere perfekte Zahlen zu bestimmen.

Beispiele:

6 = 3+2+1

28 = 14+7+4+2+1

```
perfekt?(6)=> true
perfekt?(28)=> true
```

E Rautenmuster

Schreiben Sie eine Methode **diamond**, die zu einem übergebenen ungeraden n ein Rautenmuster erzeugt. Der Rand der Raute soll als "*" gedruckt werden. Die Bereiche außerhalb der Raute als "–" und der Bereich innerhalb der Raute als Leerzeichen. Der Parameter n soll von der Konsole eingelesen werden und dann der Methode ***diamond*** als Parameter übergeben werden. Wenn n gerade oder keine ganze Zahl ist, dann soll die Methode ***diamond*** geeignete Fehlermeldungen ausgeben.

Hilfestellung: Zu einem ungeraden n betrachten wir die Mitte jeder Zeile: $n/2$. Von 0 bis $\frac{n}{2}$ ist der Spaltenindex für den linken Rand: $\frac{n}{2} - i$, für den rechten Rand: $\frac{n}{2} + i$. Von $\frac{n}{2} + 1$ bis (n-1) ist der Spaltenindex für den linken Rand: $\frac{n}{2} - (n - 1 - i)$, für den rechten Rand: $\frac{n}{2} + (n - 1 - i)$.

Beispiele:

diamond(1)

```
*
```

diamond(5)

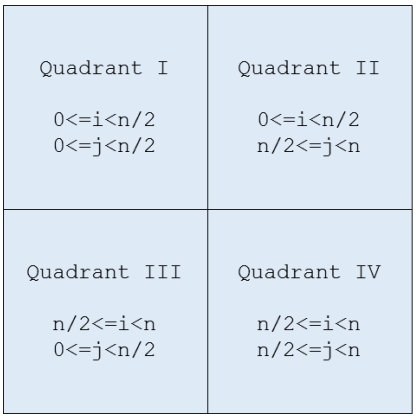
```
--*--
-* *-
*   *
-* *-
--*--
```

diamond(15)

```
-------*-------
------* *------
-----*   *-----
----*     *----
---*       *---
--*         *--
-*           *-
*             *
-*           *-
--*         *--
---*       *---
----*     *----
-----*   *-----
------* *------
-------*-------
```

F Logische Ausdrücke Auswerten

Schreiben Sie eine Methode ***truth_tables***, die die Wahrheitstabellen für AND OR, NOR und NAND erzeugt. Die Wahrheitstabelle für AND soll in Quadrant I, die für OR in Quadrant II, die für NOR in Quadrant III und die für NAND in Quadrant IV dargestellt werden. Die Bezeichnung der Quadranten und die Intervalle für die Zeilen- und Spaltenindizes sind in der nachfolgenden Abbildung gegeben.




```
i-Zeilenindex
J-Spaltenindex
```

Der logische Operator NOR ist definiert wie folgt: NOR ist ***true***, wenn keiner der beiden Operanden ***true*** ist

Der logische Operator NAND ist definiert wie folgt: NAND ist ***true***, wenn nicht beider Operanden gleichzeitig ***true*** sind.

Schreiben Sie für NOR und NAND zunächst die Wahrheitstabellen auf ein Blatt Papier.

G Notenbezeichner

Schreiben Sie eine Methode ***grades***, die Notenpunkte in Notenstufen umrechnet. Der Methode erhält als Parameter einen ganzzahligen Notenpunkt und gibt als Ergebnis die Notenstufe zurück. Wenn der Notenpunkt außerhalb des gültigen Wertebereichs liegt, dann soll ein geeigneter Fehlertext zurückgegeben werden. Verwenden Sie als Kontrollstruktur den ***case*** Ausdruck mit ***target*** und minimieren Sie die Anzahl der ***when*** Klauseln.

Zeigen Sie in einer Schleife, in der die Ergebnisse der Methode ***grades*** ausgegeben werden, dass die Methode korrekt arbeitet.

Beispiele:

grades(0) => `0 ist kein gueltiger Notenpunkt`

grades(12) => `gut`

grades(20) => `20 ist kein gueltiger Notenpunkt`

H Zufallszahlen

Ein großer Mobilfunkanbieter hat eine Marketingmaßnahme lanciert, in der die Person, die 10-mal hintereinander ein 6 würfelt, sich ein iphone 7 aussuchen kann. Sie sollen eine Methode ***the_winner_takes_it_all*** schreiben, die das Würfeln simuliert. Die Methode „würfelt“ solange, bis das erste Mal keine 6 gewürfelt wurde, oder genau 10-mal eine 6 gewürfelt wurde. Die Methode liefert als Ergebnis die Anzahl der gewürfelten 6’sen zurück. Die 10 soll der Methode als Parameter übergeben werden.

Schreiben Sie eine Schleife, die endet, wenn das erste Mal 10 6’sen in Folge gewürfelt wurden. Dann soll die Anzahl der Versuche ausgegeben werden. Lassen Sie die Schleife eine Weile laufen. Wenn Ihnen die Wartezeit zu lang wird, brechen Sie das Programm ab.

Testen Sie die Methode mit kleineren n und notieren Sie die Ergebnisse.