

Aufgabe 1

ALLGEMEINES	2
<i>Entwicklung nach Standardschema</i>	<i>2</i>
<i>Beispiele und Testfälle</i>	<i>2</i>
AUFGABEN MIT UNENDLICHER EXAKTER ARITHMETIK	3
<i>Volumen eines Hohlwürfels.....</i>	<i>3</i>
<i>Erweiterung: Volumen eines n-dimensionalen Hohlwürfels</i>	<i>3</i>
AUFGABEN MIT UNENDLICHER UNEXAKTER ARITHMETIK	4
<i>Volumen eines Zylinders</i>	<i>4</i>
<i>Oberfläche eines Pentagondodekaeders.....</i>	<i>4</i>
AUFGABEN MIT ENDLICHER EXAKTER ARITHMETIK	5
<i>Uhrenarithmetik</i>	<i>5</i>
<i>Erweiterung: Uhrenarithmetik für verschiedene Anzeigen.....</i>	<i>5</i>
<i>Freiwillige Erweiterung: Uhrenarithmetik mit Rückstellen</i>	<i>6</i>

Allgemeines

Entwicklung nach Standardschema

Entwickeln Sie die Funktionen nach unserem **Standardschema**:

1. Spezifikation (Purpose, Contract, Tests)
2. Implementation der positiven Tests
3. Implementation der Funktionsprozedur
4. Zyklus: (Refaktorisierung + Test) in kleinen Schritten
5. Implementation der Precondition-Checks
6. Testen der Checked-Function mit den positiven Tests
7. Implementation der negativen Tests
8. Testen der Funktion mit den negativen Tests

Beispiele und Testfälle

- es kommt nicht auf die **Anzahl** der Beispiele an, sondern auf ihre **Aussagekraft**
- besonders wichtig sind die **kritischen Grenzfälle**
- Testen sollte man insbesondere auch die Datenkonfigurationen, bei denen man sich seiner Implementation nicht sicher ist

Aufgaben mit unendlicher exakter Arithmetik

Volumen eines Hohlwürfels

Schreiben Sie eine Funktion, die das Volumen eines **Hohlwürfels** berechnet.

Die Kantenlängen sollen ganzzahlig sein.

Diese Aufgabe machen wir als Beispiel gemeinsam während des ersten Praktikumstermins.

Erweiterung: Volumen eines n-dimensionalen Hohlwürfels

Erweitern Sie die Funktion für das Volumen des Hohlwürfels auf n Dimensionen.

Dafür müssen Sie das notwendige zusätzliche Argument so lange weiterreichen, bis es gebraucht wird ("**Delegation**").

Aufgaben mit unendlicher unexakter Arithmetik

Volumen eines Zylinders

Schreiben Sie eine Funktion, die das Volumen eines **Zylinders** berechnet.

Dafür brauchen Sie die mathematische Konstante **Pi**. Diese ist in dem Modul Math definiert und kann als `Math::PI` referenziert werden.

Die Argumente und das Resultat sollen **Gleitkommazahlen** sein. Das bedeutet eine **andere Form der Assertions** in den Tests.

Rechnen Sie die Beispiele mit einem Taschenrechner (oder Handy oder Windows-Zubehör-Rechner) aus.

Auf jeden Fall unabhängig von Ruby. Sonst haben Sie keinen "echten" Test.

Oberfläche eines Pentagondodekaeders

Schreiben Sie eine Funktion, die die Oberfläche eines (regelmäßigen oder platonischen) Pentagondodekaeders berechnet.

Suchen Sie die Formel im Internet.

Mathematische Funktionen finden Sie im Modul Math (z.B. `sqr(x)`). Es geht aber auch ohne Math durch geeignete Verwendung der Exponentiation.

Prüfen Sie, ob Sie eine globale Konstante herausfaktorisieren können.

Aufgaben mit endlicher exakter Arithmetik

Uhrenarithmetik

Schreiben Sie eine Funktion mit der man eine übliche Digitaluhranzeige (Stunden, Minuten, Sekunden) um einen Zeitbetrag (Stunden, Minuten, Sekunden) **vorstellen** kann.

Die Uhranzeige soll 24 Stunden haben.

Das Resultat soll ein **Array** der Größe 3 sein ([hour,min,sec] z.B. [12,30,10]).

Lösungsidee: Konversion

- Konversion in Sekunden seit Anfang des Tages
- Addition mit modularer Arithmetik
- Rückkonversion in das Format [hour,min,sec]
- Definieren Sie globale Konstanten !

Erweiterung: Uhrenarithmetik für verschiedene Anzeigen

Erweitern Sie die Uhrenfunktion um ein zusätzliches Argument, das auch eine 12-Stunden-Anzeige (wie bei Analoguhren) ermöglicht.

Freiwillige Erweiterung: Uhrenarithmetik mit Rückstellen

Diese Erweiterung ist schwieriger und deshalb freiwillig.
Die Uhranzeige soll jetzt auch **zurückgestellt** werden können,
indem man negative Zeitbeträge addiert.

Überlegen Sie, was Sie sinnvollerweise zulassen wollen.

Informieren Sie sich über die Funktionsweise von **divmod** (z.B. bei ruby-doc.org). Modulare Arithmetik mit Vorzeichen ist leider etwas kompliziert. Das gilt für alle Programmiersprachen.

ALLGEMEINES	2
<i>Entwicklung nach Standardschema</i>	<i>2</i>
<i>Beispiele und Testfälle</i>	<i>2</i>
AUFGABEN MIT UNENDLICHER EXAKTER ARITHMETIK	3
<i>Volumen eines Hohlwürfels.....</i>	<i>3</i>
<i>Erweiterung: Volumen eines n-dimensionalen Hohlwürfels</i>	<i>3</i>
AUFGABEN MIT UNENDLICHER UNEXAKTER ARITHMETIK	4
<i>Volumen eines Zylinders</i>	<i>4</i>
<i>Oberfläche eines Pentagondodekaeders.....</i>	<i>4</i>
AUFGABEN MIT ENDLICHER EXAKTER ARITHMETIK	5
<i>Uhrenarithmetik</i>	<i>5</i>
<i>Erweiterung: Uhrenarithmetik für verschiedene Anzeigen.....</i>	<i>5</i>
<i>Freiwillige Erweiterung: Uhrenarithmetik mit Rückstellen</i>	<i>6</i>