

### Schiefer Wurf

#### Problemstellung:

Für eine übergeordnete Aufgabe wird der Bahnverlauf des schiefen Wurfes benötigt. Die Software soll in einer sehr einfachen Sprache realisiert werden, bei welcher noch keine Bibliotheken zur Verfügung stehen. Dadurch kann nicht auf die in Hochsprachen üblichen Befehle zurückgegriffen werden. Aus anderen Teilaufgaben stehen lediglich Funktionen für die vier Grundrechenoperationen (+ - \* /) zur Verfügung.

#### Aufgabe:

Das zu erstellende Modul soll den Koordinatenpunkt (x,y) des Wurfkörpers aus den gegebenen Grössen

- Anfangsgeschwindigkeit
- Abschusswinkel
- Zeit seit des Abwurfes

ermitteln und zurückgeben.

- a. Es soll die Struktur als Structured-Chart erstellt werden, wobei auf gute Wiederverwendbarkeit geachtet werden soll. Der Weg (x) und die Höhe (y) sollen auch individuell ermittelt werden können.
- b. Es sind die Modulspezifikationen zu den einzelnen Modulen zu schreiben.

#### Physikalische Zusammenhänge:

$v_0$  : Anfangsgeschwindigkeit

$\alpha$  : Wurfwinkel

$x$  : waagrechte Entfernung nach  $t$  Sekunden

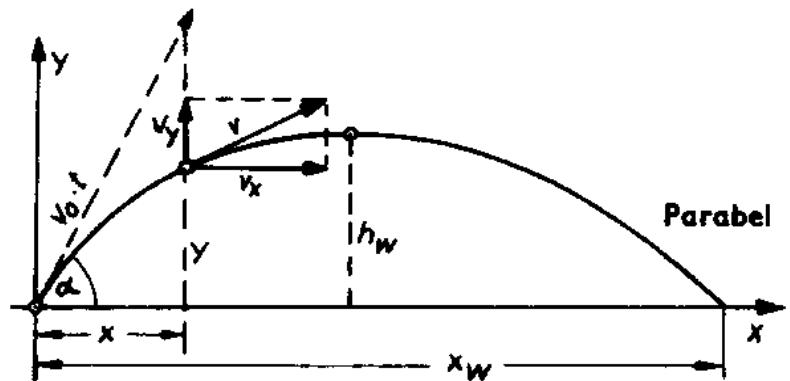
$y$  : senkrechte Entfernung nach  $t$  Sekunden

$t$  : Zeit seit des Abwurfes

$g$  : Fallbeschleunigung

$$x = v_0 \cdot t \cdot \cos(\alpha)$$

$$y = v_0 \cdot t \cdot \sin(\alpha) - \frac{g}{2} \cdot t^2$$



$$\sin(\alpha) = \frac{\alpha}{1!} - \frac{\alpha^3}{3!} + \frac{\alpha^5}{5!} - \frac{\alpha^7}{7!} + \dots \quad \alpha : \text{in rad}$$

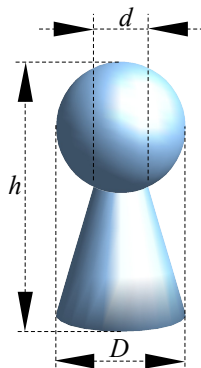
$$\cos(\alpha) = 1 - \frac{\alpha^2}{2!} + \frac{\alpha^4}{4!} - \frac{\alpha^6}{6!} + \dots \quad \alpha : \text{in rad}$$

#### Bemerkungen:

In diesem ersten Anlauf sind die Bibliothekszugriffe (Grundoperationen) nicht darzustellen.

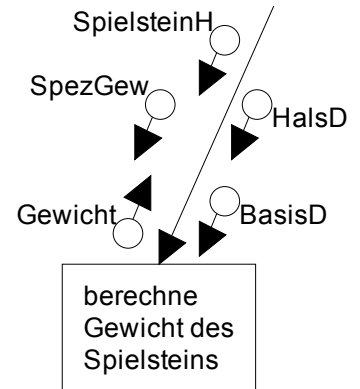
### Spielfigur

#### Problemstellung:



Das Gewicht eines Spielsteines soll durch ein Programm bestimmt werden. Der Spielstein ist aus einem Kegelstumpf und einer Kugel aufgebaut. Das Programm soll mit einem Structured-Chart strukturiert werden.

$h$  : Höhe des ganzen Spielsteines  $\rightarrow$  SpielsteinH  
 $D$  : Durchmesser der Kugel und des Fusses  $\rightarrow$  BasisD  
 $d$  : Durchmesser des Halses  $\rightarrow$  HalsD  
SpezGew : spezifisches Gewicht ( $\rho$  {roh})  
Gewicht : das ganze Gewicht des Spielsteines



#### Aufgabe:

Skizzieren Sie das angefangene Diagramm fertig.

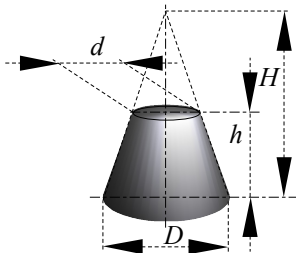
#### Hinweise:

Es ist auf gute Wiederverwendbarkeit zu achten. Es stehen nur die vier Grundrechenoperationen zur Verfügung.

#### allgemeine Formeln:

Beachten Sie, dass die Symbole der allgemeinen Formelsammlung entsprechen und nicht mit der Aufgabe korrespondieren!

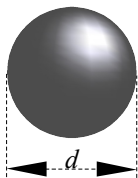
#### Kegelstumpf :



$$V = \frac{A_D \cdot H}{3} - \frac{A_d (H - h)}{3}$$

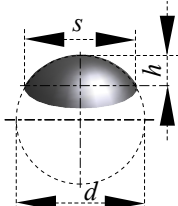
$$A_D = \frac{D^2 \cdot \pi}{4} ; A_d = \frac{d^2 \cdot \pi}{4} ; d = D \cdot \frac{H - h}{H} ; H = h \cdot \frac{D}{D - d}$$

#### Kugel :



$$V = \frac{d^3 \cdot \pi}{6}$$

#### Kugelabschnitt :



$$V = \frac{\pi \cdot h^2}{6} (3d - 2h) = \pi \cdot h \left( \frac{s^2}{8} + \frac{h^2}{6} \right)$$

$$h = \frac{d - \sqrt{d^2 - s^2}}{2}$$

### Brüche addieren

#### Problemstellung:

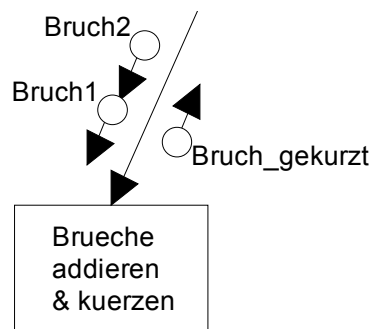
Eine Programmkomponente soll zwei Brüche addieren und das Ergebnis als Bruch gekürzt zurückgeben. Das Problem soll mit einem Structured-Chart dargestellt werden.

Eine erste grobe Modulaufteilung sieht folgende Module vor:

- „ggT“, Bibliotheksmodul für grösster gemeinsamer Teiler zweier ganzer Zahlen.
- „Brueche addieren“, addiert zwei Brüche nach dem Beispiel :  $\frac{2}{3} + \frac{4}{5} = \frac{2 \cdot 5 + 3 \cdot 4}{15} = \frac{22}{15}$  !ohne Kürzung!
- „Bruch kuerzen“, kürzt einen Bruch z.B. :  $\frac{9}{15} = \frac{3}{5}$

#### Aufgaben:

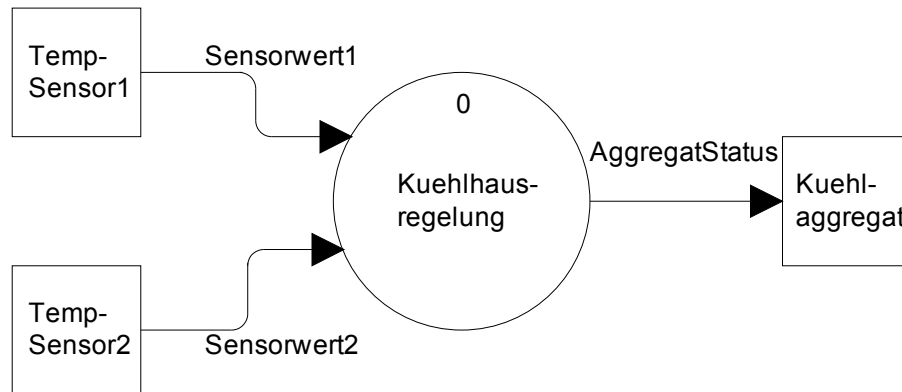
1. Zeichnen Sie das Structured-Chart fertig! (Bem: wenn sinnvoll können weitere Module eingefügt werden.)
2. Erstellen Sie die Modulspezifikation des Moduls „ggT“. Ueberlegen Sie sich dabei auch sinnvolle Rückgabewerte, wenn eine oder beide Zahlen 0 sind.



## Kühlraumregelung „erstellen“

### Problemstellung:

Ein Kühlraum soll auf konstanter Temperatur gehalten werden. Der Raum verfügt über zwei unabhängige Temperatursensoren welche einen von der Temperatur abhängigen Widerstandswert liefern. Die Sensoren werden regelmässig abgefragt und die beiden errechneten Temperaturen gemittelt. Liefern die beiden Sensoren aber zu unterschiedliche Temperaturwerte wird ein Alarm ausgelöst. Damit sich sprunghafte Aenderungen nicht sofort auswirken werden jeweils die letzten 4 gemittelten Temperaturwerte erneut gemittelt. Dieser Wert entscheidet dann, ob das Kühlaggregat ein- oder ausgeschaltet wird.



Datenflussdiagramm „Kontext“

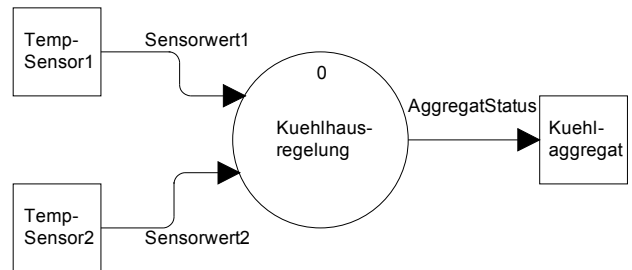
### Aufgabe:

Erstellen Sie das Structured Chart für diese Regelung!

### Kühlraumregelung „beurteilen“

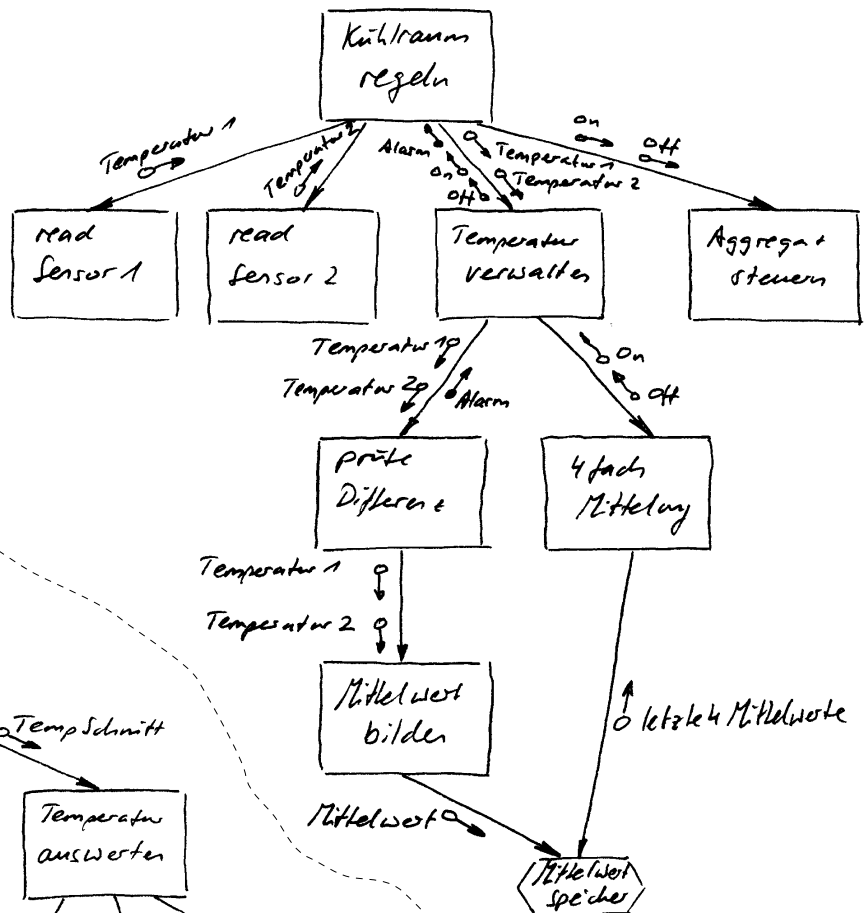
#### Problemstellung:

Ein Kühlraum soll auf konstanter Temperatur gehalten werden. Der Raum verfügt über zwei unabhängige Temperatursensoren welche einen von der Temperatur abhängigen Widerstandswert liefern. Die Sensoren werden regelmässig abgefragt und die beiden errechneten Temperaturen gemittelt. Liefern die beiden Sensoren aber zu unterschiedliche Temperaturwerte wird ein Alarm ausgelöst. Damit sich sprunghafte Aenderungen nicht sofort auswirken werden jeweils die letzten 4 gemittelten Temperaturwerte erneut gemittelt. Dieser Wert entscheidet dann, ob das Kühlaggregat ein- oder ausgeschaltet wird.

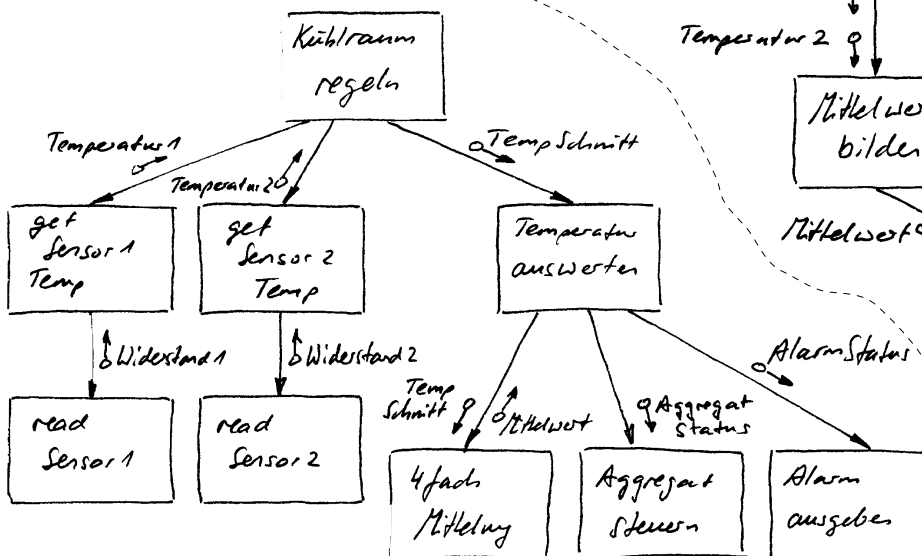


#### Aufgabe:

- Zu der Problemstellung wurden zwei Vorschläge gemacht. Beurteilen Sie diese Vorschläge und notieren Sie was daran gut, falsch, problematisch etc. ist.
- Versuchen Sie nach Möglichkeit eine bessere Struktur zu finden.



Structured-Chart 1



Structured-Chart 2

## Text raten

### Problemstellung:

Beim bekannten Fernsehspiel „Glücksrad“ mussten Begriffe oder ganze Sätze erraten werden. Eine vereinfachte Version soll mit einem Programm realisiert werden.

### Vorgaben:

Das zu erstellende Programm fragt einen Satz ab, der durch eine Zweitperson einzugeben ist. Der Spieler sieht statt der Buchstaben nur Fragezeichen. Satz- und Leerzeichen werden normal dargestellt. Nach der Eingabe eines Buchstaben wird dieser an die Stelle eines bzw. mehrerer Fragezeichen gesetzt, sofern der Buchstabe in dem zu ratenden Satz vorkommt. (Gross-/Kleinschreibung soll nicht beachtet werden). Für jedes Vorkommen eines erratenen Buchstaben gibt es einen Punkt. Wird in einem Versuch kein Buchstabe erraten, wird ein Punkt abgezogen.

Das Programm endet, wenn der Satz geraten oder eine vorgegebene Zahl an Versuchen erreicht wurde.

Erste Überlegungen haben zu folgender Lösungsidee geführt.

- Es wird nur der 7-bit ASCII-Code erlaubt
- Die Buchstaben werden im Satz verdeckt, indem diese speziell verschlüsselt werden. Die Verschlüsselung wird so realisiert, dass erkennbar ist, ob ein Zeichen normal oder verschlüsselt ist.
- Verschlüsselte Zeichen werden als Fragezeichen ausgegeben
- Bei korrekt erratenen Buchstaben wird das verschlüsselte Zeichen durch den erratenen Buchstaben ersetzt
- Der Satz wurde korrekt erraten, wenn kein Zeichen mehr verschlüsselt wurde.

Daraus wurden bereits zwei Funktionen realisiert, welche zur Verfügung stehen:

<b>encryptChar( Zeichen )</b>	Diese Funktion verschlüsselt das Zeichen und gibt es als Returnwert zurück.
<b>checkEncryption( Zeichen )</b>	Diese Funktion prüft, ob das Zeichen verschlüsselt ist und gibt in diesem Fall TRUE, sonst FALSE zurück.

### Aufgaben:

- a) Erstellen Sie das Structured-Chart unter Einbezug der bereits existierenden Funktionen. Der eingegebene Satz und die Punktzahl sollen als lokale Variablen im Hauptprogramm gehalten werden.
- b) Erstellen Sie die Modulspezifikationen aller Module
- c) Erstellen Sie zu den komplexeren Modulen auch Struktogramme

### Variante:

Um den Einstieg etwas zu erleichtern, kann ein Buchstabe zufällig oder durch Positionsangabe des Spielers zu Beginn aufgedeckt werden.