



F 1 Signatur von Funktionen

I. Vorstellung einer Maschine

Variablen

x1, x2, x3



Maschine
(Black Box)

Variable

y

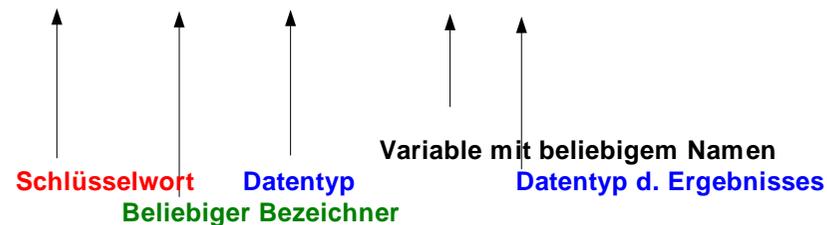


F 1 Signatur von Funktionen

II. Signaturschreibweise

Signatur einer Funktion:

```
function Addition (int x1, int x2): int
function Addition2 (float x1, float x2): float
```



F 1 Signatur von Funktionen

Aufgaben

Gib die Signaturen an:

- Größter gemeinsamer Teiler (ggT)
- Fakultät ($5! = 120$ weil $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$)
- Lösung x1 einer quadratischen Gleichung
- Wandle einen Buchstaben in den ASCII-Code um
- Abstand zweier Punkte voneinander



F 2 Variablen

Variablen

Variable

Schachtel
mit Namen
z.B.
\$Name



Wertzuweisung

\$Name="Huber"



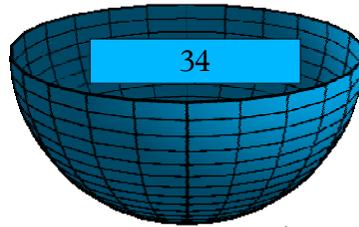


F 2 Variablen

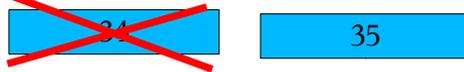
Variablen

Variable

Schachtel
mit Namen
z.B.
\$Alter



Wertzweisung ~~\$Alter=\$Alter+1~~



F 2 Variablen

Aufgaben

Gib den End-Wert der Variablen an:

- \$Name="Meier"; \$Name="Schulze"; \$Name="Huber";
- \$i=17; \$i=\$i-7; \$i=\$i*10;
- \$i=5; \$j=7; \$k=\$i+\$j*\$k;
- \$netto=230; \$Artikelanzahl=10; \$MWST=0.16;
\$Ergebnis=\$netto*\$Artikelanzahl*\$MWST;
- Strings werden mit einem "." konkateniert
\$Name="Huber"; \$Vorname="Herbert";
\$Gesamt=\$Vorname." ".\$Name;



F 3 Funktionen

schon vorhandene Funktionen (sehr kleine Auswahl)

Folgende Funktionen sind in PHP schon vorhanden:

ord(String[1] A):int	// ermittelt ASCII-Code
chr(int A): String[1]	// ermittelt das ASCII-Zeichen
sqrt(float): float	// Quadratwurzel
pow(int Wert1, int Wert2):Int	// Potenz Wert1 hoch Wert2
abs(float Wert):float	// ergibt absoluten Wert
strlen(String Name):int	// ergibt die Länge eines Strings
\$Name[1]:char	// ergibt das Zeichen an erster Stelle

Operatoren

Neben +,-,/ und * gibt es den Operator % (Modulo), der den Rest einer Ganzzahlendivision berechnet!

Bsp: \$x=10 % 3; echo \$x;



F 4 Eigene Funktionen

Aufbau einer Funktion

Aufbau einer Funktion:

```
function Addition($zahl1, $zahl2){
    return $zahl1 + $zahl2;
}
```



F 4 Eigene Funktionen

Beispiel

```
function GetLaengePunkte($x1,$y1,$x2,$y2){  
    return sqrt(abs($x1-$x2)*abs($x1-$x2)+abs($y1-$y2)*abs($y1-$y2));  
}
```



F 4 Eigene Funktionen

Aufgaben

1. Schreibe zwei Funktionen für die Nullstellenberechnung

```
function GetX1(float $a, float $b,float $c): float  
function GetX2(float $a, float $b,float $c): float
```

2. Schreibe eine Funktion zur Berechnung der Sekunden, die seit Mitternacht vergangen sind

```
function GetSek(int $std, int $min): int
```



F 4 Eigene Funktionen

Aufgaben

3. Schreibe eine Funktion zur Berechnung der laufenden Nummer eines Feldes im Schachbrett, wenn nur die Koordinaten in x- und y-Richtung (Werte von 1 bis 8) bekannt sind

```
function GetNr(int x, int y): int
```

4. Schreibe eine Funktion, die einen Namen in eine beliebige aber (pseudo-)zufällige Zahl zwischen 1 und 15 konvertiert.

Hinweis: `$x=$String[1]` ergibt das erste Zeichen der Variablen;

```
function GetWert(String $Name): int
```



F 5 Algorithmus

Definition

Def.: Algorithmus: Eine Verarbeitungsvorschrift, die von einer Maschine verstanden wird. Die Anweisung muss eindeutig sein, kann aber Wahlmöglichkeiten oder z.B. auch Zufallszahlen beinhalten.

Ein Algorithmus gibt an, wie Eingabedaten schrittweise in Ausgabedaten umgewandelt werden.

Euklid hat schon 300 v.Chr. den Algorithmus für die Bestimmung des ggT angegeben!





F 5 Algorithmus

Definition

```
ggT(7,12)
= ggT(7,5)
= ggT(2,5)
= ggT(2,3)
= ggT(2,1)
= ggT(1,1)
BREAK -> Ergebnis: 1
```

Ziehe solange von der größeren Zahl die kleinere ab, bis die beiden Werte gleich sind, dann hast du das Ergebnis!

Berechne mit diesem Algorithmus den
ggT(63,9)



F 5 Algorithmus

Definition

```
ggT(63,9)
= ggT(54,9)
= ggT(45,9)
= ggT(36,9)
= ggT(27,9)
= ggT(18,9)
= ggT(9,9)
BREAK -> Ergebnis: 9
```

Ziehe solange von der größeren Zahl die kleinere ab, bis die beiden Werte gleich sind, dann hast du das Ergebnis!

```
ggT(6,12)
= ggT(6,6)
BREAK -> Ergebnis: 6
```



F 5 Algorithmus

Umsetzung in PHP

```
function ggT(int a, int b):int
```

in PHP:

```
function ggT($a,$b){
    ... return $a else ...
}
```

Die if-Anweisung erlaubt in PHP eine Verzweigung der Anweisung:

```
if ($Alter > 20) echo
    "Sie sind ein alter Sack";
else echo
    "Hi Du!";
```

Schreibe eine Funktion in PHP, die den ggT berechnet!



F 5 Algorithmus

Umsetzung in PHP

**Schlüsselwort
"function"**

Bezeichner beliebig

**Variablen, die oben
in die Maschine
eingeworfen werden**

Ergebnis

Anweisungsblöcke { }

```
function Addition($a,$b){
    return $a+$b;
}
```



F 6 Abstraktion

Abstraktion ist ein wichtiges Prinzip der Informatik

Ein Gesamtproblem ist komplex!

**Aufteilen des Problems in kleine Einheiten,
die übersichtlich sind.**

(Für jedes kleine Problem eine Funktion.)

**Zusammensetzen der Teilprobleme zur
Gesamtlösung.**

Bsp.: Autoherstellung **Auto(Motor, Türen, Räder, ...)**
 Motor(Kühlung, Zylinder, ...)
 Zylinder(Zündkerze, Kolben...)
 Zündkerze(Spule, Kontakte...)
 ...



F 6 Abstraktion

Abstraktion ist ein wichtiges Prinzip der Informatik

**Verbindendes Element der Teillösungen
sind die Signaturen.**

**Auf einer höheren Abstraktions-Ebene muss der
Programmierer nicht wissen, wie das Problem gelöst
wird, er muss es einfach verwenden können (Black-
Box-Sicht).**



F 6 Abstraktion

Abstraktion ist ein wichtiges Prinzip der Informatik

Beispiel: Ein Hubschrauberpilot benötigt eine Software zur
Berechnung von Flugrouten

**Nach Eingabe von Start- und Zielkoordinaten, der
Geschwindigkeit in km/h sowie der Ankunftszeit soll die
Software berechnen, wann und in welche Richtung er
losfliegen muss, um das Ziel rechtzeitig zu erreichen. Auch
der Weg (Luftlinie) soll angegeben werden.**

**Überlege dir, wie sich die Aufgabe in einzelne Teilaufgaben
zerlegen lässt!**
**Eine Möglichkeit der Implementierung besteht in folgenden
Signaturen:**



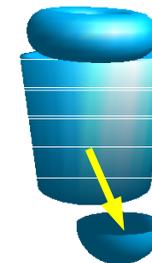
F 6 Abstraktion

Das Gesamtproblem ist sehr komplex!

Beispiel:

Std, Min, x1,y1,x2,y2,v

10:20 Uhr, Start: 5,5 Ziel: 10,5,v=10 km/h



**Maschine
(Black Box)**

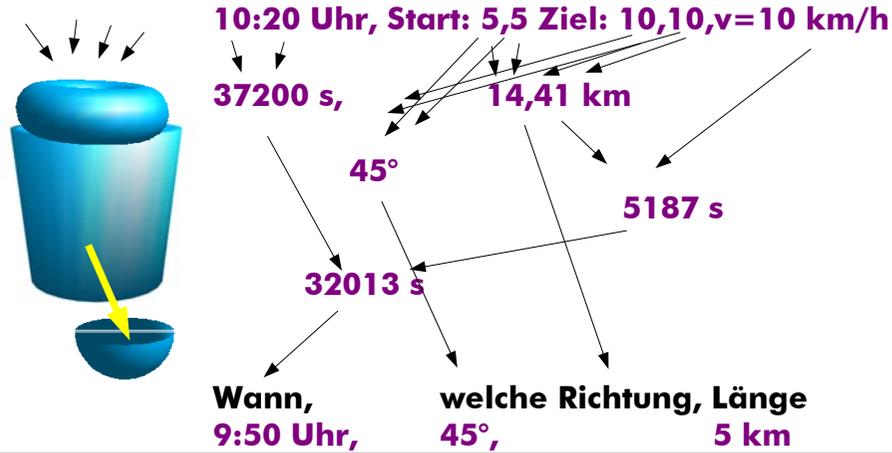
**Wann, welche Richtung, Länge
9:50 Uhr, 90°, 5 km**



F 6 Abstraktion

Suche nach Teilproblemen (top-down-Ansatz)

Beispiel:



F 6 Abstraktion

Aufteilung in Teilprobleme "Teile und Herrsche"

Funktion 1

```
// Ermittlung der Sekunden aus der Eingabe von
// Stunde und Minute
```

```
function getSek($h,$m):int
```



F 6 Abstraktion

Aufteilung in Teilprobleme "Teile und Herrsche"

Funktion 2

```
// Ermittlung eines Strings der Sekunden in eine Uhrzeit
//ausgibt
```

```
function getUhrzeit($sek):String
```



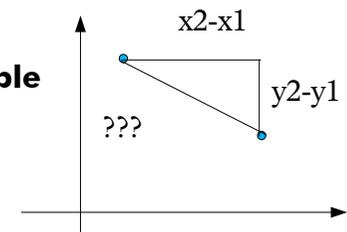
F 6 Abstraktion

Aufteilung in Teilprobleme "Teile und Herrsche"

Funktion 3

```
// Ermitteln des Weges zwischen Start und Ziel
// $x1,$y1 sind die Startkoordinaten
// $x2,$y2 sind die Zielkoordinaten
// Berechnung über den Pythagoras
```

```
function getWeg($x1,$y1,$x2,$y2):double
```





F 6 Abstraktion

Aufteilung in Teilprobleme "Teile und Herrsche"

Funktion 4

```
// Ermittlung der Startzeit  
// Eingegeben wird die Zielzeit und die Flugzeit  
// beide Werte in Sekunden seit Mitternacht
```

```
function getStartzeit($zielzeit,$flugzeit):double
```



F 6 Abstraktion

Aufteilung in Teilprobleme "Teile und Herrsche"

Funktion 5

```
// Ermittlung des Winkels  
// Koordinaten wie oben, über Arcus-Sinus-Funktion
```

```
function getWinkel($x1,$y1,$x2,$y2):float
```