

Programmierung

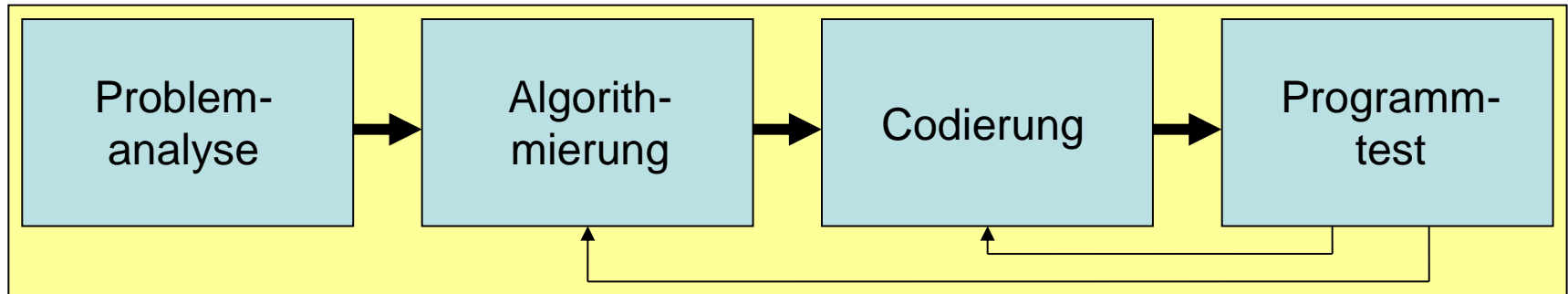
Was bedeutet Programmieren ?

Welche Programmiersprache ist geeignet ?

Welche Regeln sollen/müssen bei der Programmierung beachtet werden ?



Vom Problem (Aufgabe) zum Programm



- Was soll erreicht werden ?
- Welchen Daten müssen erfasst werden ?
- Welche Hilfsmittel stehen ggf. zur Verfügung ?

- Aufstellen einer allg. Schrittfolge zur Lösung des Problems
- Strukturierung der Daten
- **PAP**
- **Struktogramm**

- Umsetzung des Algorithmus in die Befehle einer (geeigneten) **Programmiersprache**
- **Quelltext (Code)**

- Überprüfung der korrekten Arbeitsweise an Hand von Beispieldaten
- **ggf. Korrektur des Algorithmus bzw. Quelltextes**

Unter **Programmierung** versteht man den Prozess der algorithmischen Aufbereitung und Umsetzung eines Problemlösungsprozesses am Computers.

Ein **Computerprogramm** beschreibt einen Algorithmus mit den Mitteln einer **Programmiersprache**.

Programmiersprachen:

Eine **Programmiersprache** ist ein künstliches Sprachsystem zur Beschreibung von Algorithmen und Datenstrukturen mit festen Regeln (*Syntax, Semantik*).

Jede Programmiersprache hat ihren spezifischen Anwendungsbereich (Berechnungen, Grafik/Animation, WEB, ...)

Programmiersprachen werden nach verschiedenen Kriterien klassifiziert:

systemorientierte
Programmiersprachen

z.B.

höhere
Programmiersprachen

- Maschinensprache:

01100100 11100010 ...

- Assemblersprache

mnemonische Befehle: add A, 0

- sind der menschlichen
Sprache angepasst

- müssen vor ihrer Ausführung
In die Maschinensprache
übersetzt werden

Bei den höheren Programmiersprachen unterscheidet man verschiedene Programmierparadigmen (*Programmierstile*).

- imperative, prozedurale, deklarative Sprachen
- objektorientierte Sprachen
- ...

Wahl der Programmiersprache und Programmierumgebung ?

```
// Include needed header files
#include <OC_const.h> // OC Constants
#include <Common.h> // Basic types
#include <stdio.h> // Printf
#include <Data.h> // Vectors and matrices
#include <OC_neg8.h> // NAG function

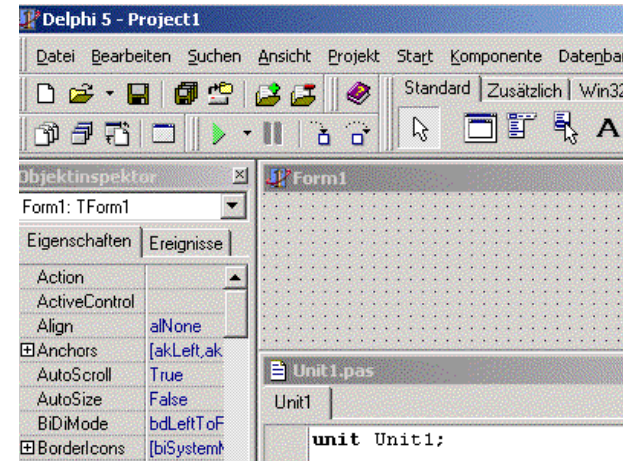
int GetData()
{
    // Declare local variable holding error code
    int iErr;

    // *** Declare and initialize input parameters ***
    // *** passed into the NAG function nag_regns_nult_linear ***
    int nPts; // Input number of rows
    int nTdx; // Input number of columns
    matrix mX; // Input matrix containing data points
    vector vY; // Input vector containing data points
    vector vWT; // Input vector containing weights

    // There are 10 rows of data in sample data
    nPts = 10;

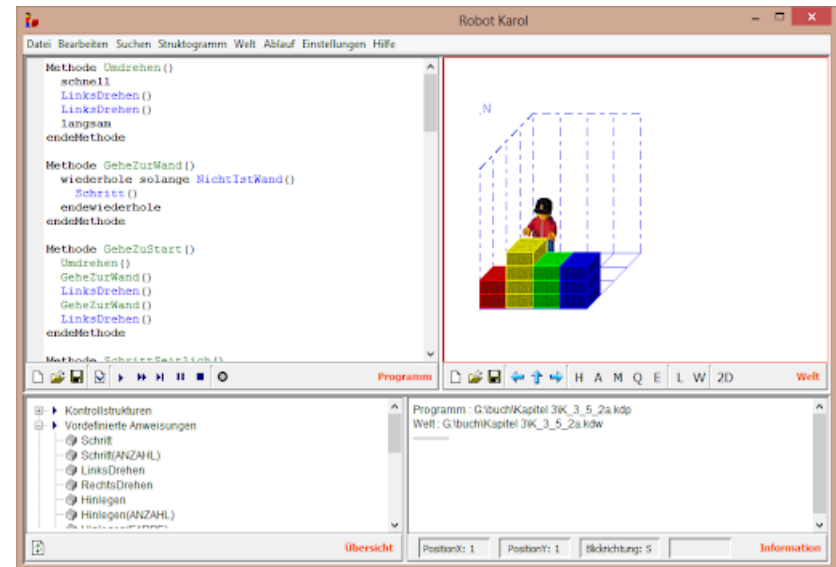
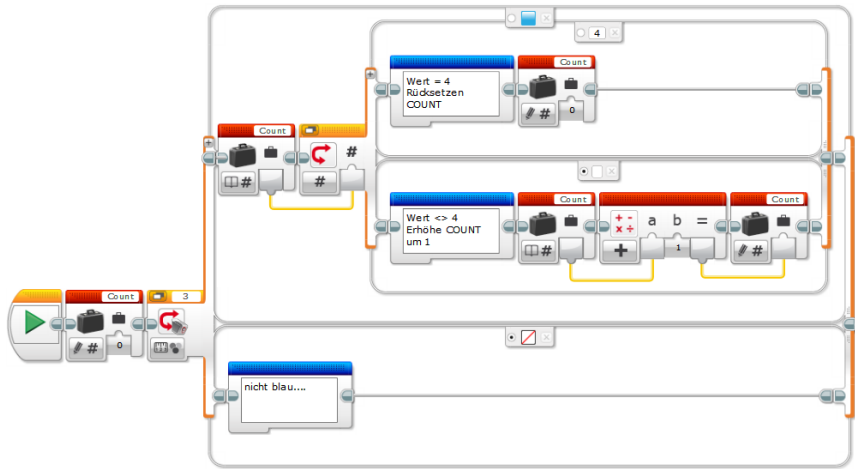
    // There are 3 independent variables in data
    nTdx = 3;

    // Get independent variable data sets
    Worksheet wks("Data1"); // Attach OC Worksheet object
    mX.CopyFromWks( wks, 1, 3, 0, 9); // Copy independent variables
}
```



... reine Textdarstellung (Java)

... in Objektdarstellung (VB-Script)



... mit Programmblöcken (LEGO)

... visuelle Programmierung

Die Programmiersprache **Python**

- wurde Anfang der 1990er Jahre entwickelt (2000, 2008 grundlegend erweitert)
- universelle, höhere Programmiersprache.
 - einfache Berechnungen
 - grafische Programmierung
 - WEB-Programmierung
 - Datenbanken, Spiele, ...
- einsetzbar für komplexe Softwareprojekte
- unabhängig vom Betriebssystem
- freie, unlizenzierte Programmiersprache
- einfache Lesbarkeit und Struktur
- i.R. wenige Codezeilen
- unterstützt verschiedene Programmierparadigmen
- Pythonmodule können in anderen Programmiersprachen eingebunden werden
- ständige Weiterentwicklung der Programmiersprache und große Community
- ▶ aktuelle Version: 3.7 ... auch in deutscher Sprache ☺



python™

*... nach Monty Python
(britische Komikergruppe)*



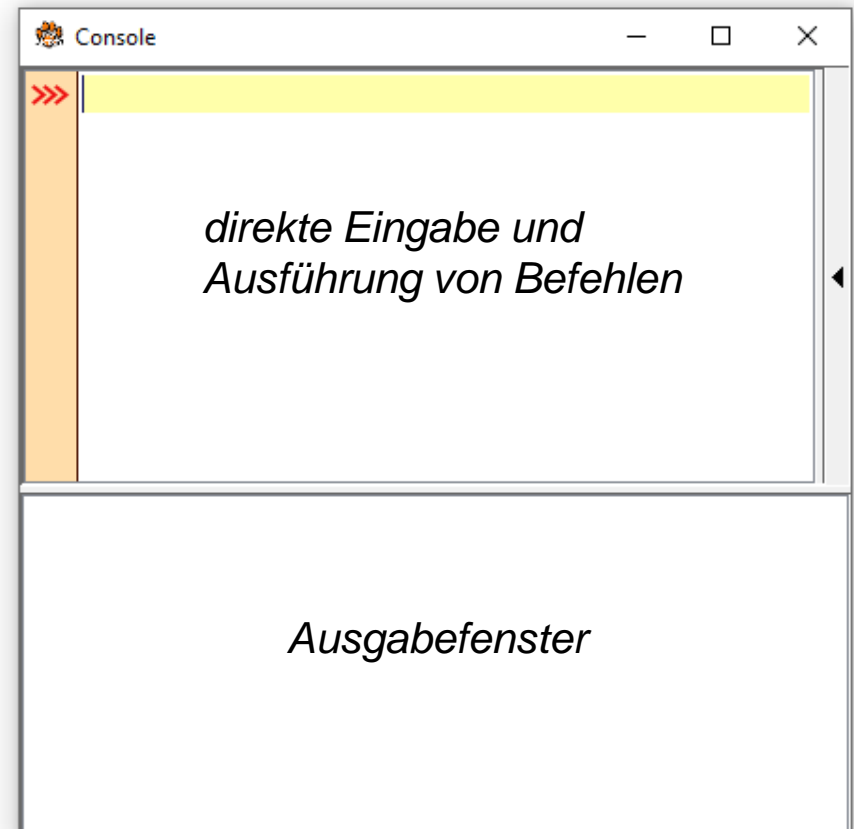
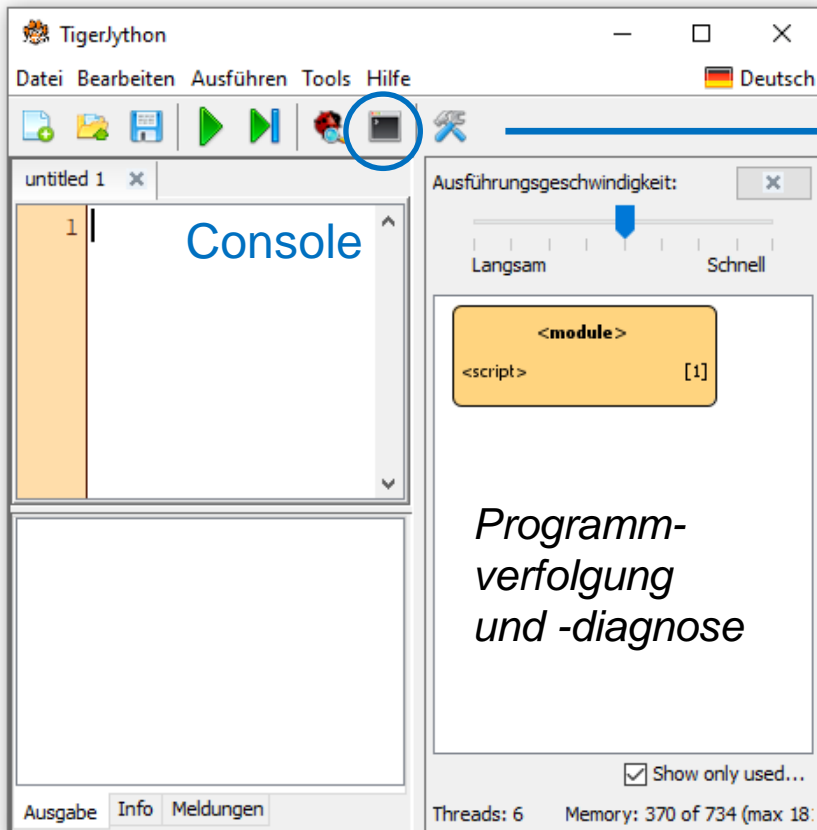
**Python ist gut für
einen Einstieg in die
Programmierung
geeignet**

Arbeit mit TigerJython

Jython ist eine Python-Implementierung mit Java als Lehrmittel für den Unterricht



Programmoberfläche

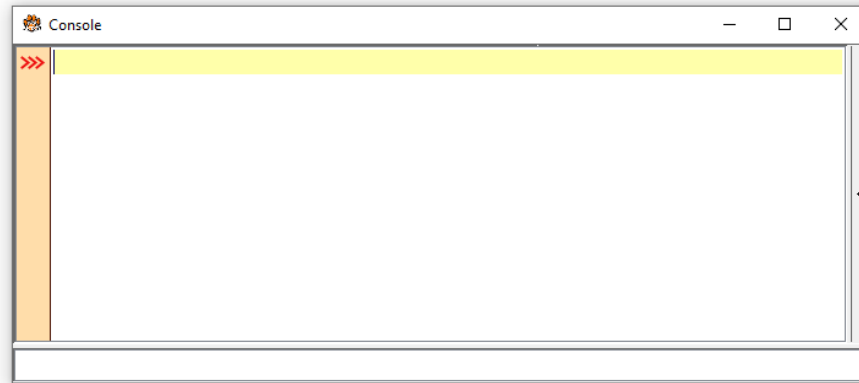


Nutzung der Console (Befehlszeile):

Die Console (Shell) dient der direkten Eingabe von Anweisungen, welche nach Abschluss (Enter) übersetzt (interpretiert) und ausgeführt werden.

Beispiel:

*... einfache
mathematische
Berechnungen ...*



Grundrechenoperationen: + , - , * , /

→ *man kann mit ganzen Zahlen (integer) und gebrochenen Zahlen mit Dezimalpunkt (float) arbeiten*

→ *automatische Vorrangrechnung*

Ganzzahl-Division: // 5 // 3 → 1

Modulo Division: % 5 % 3 → 2 ... *ganzzahliger Rest*

Potenzieren: **

► **Komplexere Berechnungen erfolgen mit Funktionen.**

Allgemeingültigkeit ...

Mit Hilfe von **Variablen** kann in Programmen mit beliebigen Zahlenwerten (und Texten) gearbeitet werden.

Variablen werden mit Hilfe eines **Bezeichners** (Namen) festgelegt.

Der Bezeichner muss aus mindestens einem Buchstaben bestehen und darf keine Leerzeichen, Umlaute, Akzente und Sonderzeichen einhalten.

Jeder Variablen ist ein Datentyp zugeordnet.

| <u>numerisch</u> | | <u>Text</u> |
|-------------------------------------|--|---|
| <u>integer</u> | <u>float</u> | Zeichenketten (string) |
| vorzeichenbehaftete ganze Zahlen | Kommazahlen mit <u>Dezimalpunkt</u> | als Folge von Zeichen in Anführungszeichen " ... " |

Eine Wertzuweisung an eine Variable erfolgt mit dem Zuweisungsoperator = (*Gleichheitszeichen*).

Bezeichner = Wert

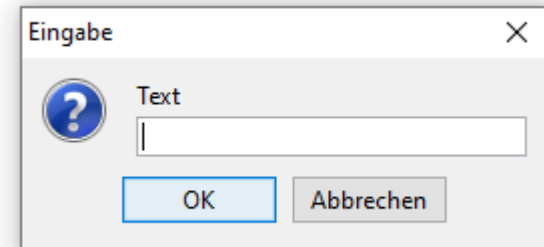
z.B.: Zahl1 = 123
 Zahl2 = 1.23

Text1 = "Hallo"
Text2 = "123"

EVA – Prinzip:

Eingabe: Bezeichner = `input("Text")`

Die Eingabe erfolgt in einer Dialogbox mit dem Aufforderungstext "Text".



Der Wert wird in der Variablen mit dem Namen Bezeichner gespeichert.

Es erfolgt eine automatische Erkennung des Variablentyps !!!

Ausgabe: `print (Bezeichner/Wert/Text)`

Die Textausgabe erfolgt mit: `print ("Text")`

Ausgabe mehrerer Texte: `print ("Text1" + "Text2")`

Mehrfachausgabe von Text: `print ("Text" * faktor)`

Zeilenwechsel: `print ("Text1" + "\n" + "Text2")`

Ausgabe von Text und
numerischer Variablen: `print ("Text1" , Variable)`

```
print("Guten Tag")
name=input("Wie heißt du ?")
print("Hallo "+name)
print("Ich möchte mit dir rechnen !")
a=input("Gib mir eine erste Zahl")
b=input("Gib mir noch eine zweite Zahl")
s=a+b
d=a-b
p=a*b
q=a/b
g=a//b
m=a%b
p=a**b
print('Die Summe beträgt: ', s)
print("Die Differenz beträgt: ", d)
print("Das Produkt beträgt: ", p)
print("Der Quotient beträgt: ", q)
print("Ganzzahldivision:: ", g)
print("Modulodivision: ", m)
print("Die Potent beträgt: ", p)
```