

LANDESBERUFSSCHULE 4 SALZBURG

# Grundbegriffe der Informatik

---

Version 1.3

Dieses Skript dient als zusätzliche Lernunterlage für Informatik

**Inhalt**

Informatik.....	4
Software .....	4
Korrektheit .....	4
Benutzbarkeit .....	4
Wartbarkeit .....	4
Effizienz.....	4
Begriffe der Informatik (allgemein).....	5
Informatik .....	5
EDV .....	5
Computer .....	5
Begriffe der Informatik (programmiertechnisch).....	5
EVA-Prinzip .....	5
Entwicklungsumgebung (IDE) .....	6
Quellcode .....	6
Programmiersprachen .....	6
Was sind Programmiersprachen .....	6
Typen von Programmiersprachen .....	6
Maschinensprachen .....	6
Assemblersprachen.....	7
Hochsprachen .....	7
Datentypen, Variablen, Konstanten .....	7
Variable .....	7
Konstante .....	7
Datentyp.....	7
Datenstruktur.....	8
Anweisungen .....	8
Deklarieren .....	8
Initialisieren .....	8
Kontrollstrukturen .....	8
Kopfgesteuerte Schleife .....	8
Fußgesteuerte Schleife .....	8
Zählschleife .....	8
Bedingte Anweisung und Verzweigung .....	8
IF-ELSE.....	8
SWITCH-CASE .....	8
Syntaktische Begriffe .....	9
Programmiersprache.....	9

Algorithmus .....	9
Syntax .....	9
Programm .....	9
Prozess .....	9
Vom Quelltext zum ausführbaren Programm.....	10
Interpreter .....	10
Compiler .....	10
Übung 1 .....	11
Übung 2.....	11
Anmerkung: .....	11

## Grundbegriffe

## Informatik

Das Wort Informatik setzt sich aus den Wörtern „Informationen und Automatik“ zusammen. Informatik wird sehr oft mit Programmieren gleichgestellt. Das stimmt nur bedingt, weil die Informatik die ganze Breite der Informationsverarbeitung abdeckt. Sie teilt sich in drei Hauptgruppen.



Die Wurzeln liegen in der Mathematik. Das Ziel war damals die Automatisierung von Berechnungen. Im nächsten Schritt wurden logische Maschinen entwickelt die selbständig Entscheidungen treffen konnten. Das verschmelzen der Technologien führte Schritt für Schritt zur derzeitigen Stand der Informationstechnologie.

Sie werden sich in diesem Fach hauptsächlich mit der praktischen Informatik beschäftigen. Programmieren ist eine sehr kreative Tätigkeit. Das bedeutet, dass der Softwareentwickler die Fähigkeit zur Problemlösung besitzen soll. Problemlösung wird in diesem Zusammenhang mit der Entwicklung von Algorithmen gesehen.

Programmierung bedeutet die Umsetzung eines Softwareentwurfs in Quellcode. Der Softwareentwickler hat aber noch andere Aufgaben, wie die Modellierung eines Konzeptes und das Testen der Programme.

**Software** unterliegt gewissen Qualitätskriterien, damit diese auch verlässlich funktioniert.

### Korrektheit

- Das Programm muss fehlerfrei sein und genau das tun was erwartet wird

### Benutzbarkeit

- Verständlicher Aufbau der Menüs
- logische Bedienbarkeit
- Standardisierung (Design)

### Wartbarkeit

- Änderungen und Anpassbarkeit sollen einfach zu realisieren sein
- Verwendung von selbsterklärenden Variablen und Kommentaren
- Portabilität (Verwendung auf mehreren Plattformen)

### Effizienz

- Möglichst wenig Ressourcenverbrauch ( Speicher, Bandbreite, Laufzeit)

## Begriffe der Informatik (allgemein)

Die nachstehenden Begriffe sollte jeder kennen und erklären können.

### Informatik

Ist die Wissenschaft der systematischen Verarbeitung von Informationen, besonders die automatisierte Verarbeitung mit Unterstützung von Digitalrechnern.

### EDV

Sammelbegriff für die Erfassung und Verarbeitung von Informationen (Daten) durch elektronische Maschinen. Umfasst die Datenverarbeitung und Datenverarbeitungssysteme

### Computer

Ein elektrisches (elektronisches) Gerät, das mittels programmierbarer Rechenverfahren Daten verarbeitet. Computer speichern und verarbeiten Daten. Dazu benötigt der Computer

Hardware: maschinentechnische, physikalische Ausstattung eines Computers (Prozessor, Speicher, Ein-/Ausgabegeräte)

Software: Programme bestehen aus einer Abfolge von Anweisungen/Befehlen (Instruktion), die die Hardware steuern. Ohne Programme (Software) ist ein Computer eine „leere“ Maschine.

## Begriffe der Informatik (programmiertechnisch)

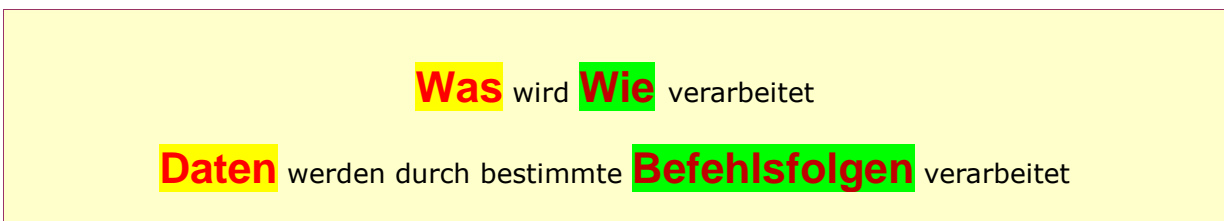
### EVA-Prinzip

Das EVA-Prinzip beschreibt ein Grundprinzip der Datenverarbeitung. Die Abkürzung bedeutet

**EINGABE → VERARBEITUNG → AUSGABE**

und beschreibt die Reihenfolge, wie die Daten verarbeitet werden.

Wenn man die Verarbeitung betrachtet ist folgendes bedeutsam:



Die Eingaben können räumlich oder zeitlich gesehen werden.

räumlich: Eingabe mit Maus, Tastatur, Scanner, andere Systeme, ...

zeitlich: Alle Eingaben erfassen (keine weiteren möglich), danach verarbeiten

Ausgaben können Monitor, Drucker, Signallampen, usw. sein

### Entwicklungsumgebung (IDE)

Programmsammlung mit der es möglich ist Softwareentwicklung ohne weitere Anwendungen zu erzeugen. Sie besteht mindestens aus

- Texteditor
- Compiler
- Linker
- Debugger
- Hilfesystem

Weitere Funktionen für Versionsverwaltung, Projektmanagement usw. können ebenfalls enthalten sein.

Beispiele für IDE sind Eclipse, Visual Studio Express, DevCpp

### Quellcode

Menschenlesbarer Code (Text), der in einer Programmiersprache geschrieben wurde

## Programmierersprachen

### Was sind Programmiersprachen

- „ähneln einer Fremdsprache“
- „bestehen aus Zeichen und Regeln und dienen der Kommunikation mit dem Computer“
- „ermöglichen die Erstellung von Programmen“
  
- „Eine Programmiersprache ist eine formale Sprache zur Darstellung von Computerprogrammen.“
- „Eine logische Abfolge von Befehlen in einer Programmiersprache nennt man allgemein Computerprogramm

### Typen von Programmiersprachen

Es gibt drei grundsätzliche Typen an Programmiersprachen:

- 1.Maschinensprachen
- 2.Assemblersprachen
- 3.Hochsprachen

### Maschinensprachen

- einzige Sprache die direkt vom Computer verstanden wird
- jeder Befehl wird in Zahlencodes dargestellt z.B.: 10 → addieren
- für Menschen mühsam zu interpretieren

### Assemblersprachen

- sind Maschinennahe Programmiersprachen
- sind spezifisch oder auf wenige Prozessoren ausgerichtet
- performante Programmierung möglich  
z.B.: `mov ax,bx`

### Hochsprachen

- benutzen textuelle- und mathematische Notation
- Befehle müssen übersetzt werden
- Erleichtern das Programmieren von Maschinen
- zwei Arten
  - Compiler-Sprache
  - Interpreter-Sprache

## Datentypen, Variablen, Konstanten

Die zu verarbeiteten Daten werden im Hauptspeicher (RAM) gespeichert. Speicher sind in Bit, Byte und Wort aufgeteilt.

Wir wissen, dass die elementare Informationseinheit ein Bit ist. 8 Bits ergeben ein Byte. Und die Registerbreite der verwendeten CPU bestimmt die Breite eines Speicherwortes.

Der gesamte Speicher wird vom Betriebssystem aufgeteilt und jede Zelle erhält eine Adresse. Die ProgrammiererInnen müssen sich die Adressen nicht merken. Sie können stattdessen Namen verwenden.

### Variable

Platzhalter für Speicherplatz. Dieser wird durch den Datentyp bestimmt. Der Name kann frei gewählt werden, sollte aber sprechende Namen besitzen. Variablen können im Verlauf des Programmes verschiedene Werte oder Datentypen annehmen. Der Wert kann sich während der Laufzeit ändern.

Der Name einer bestimmten Variable sollte immer aussagekräftig sein.

### Konstante

Datentyp deren Wert sich während der Laufzeit nicht ändern kann

### Datentyp

Wird in der Programmierung verwendet um Speicherbereiche eine konkrete Semantik zuzuweisen. Es gibt drei verschiedene Arten von Datentypen:

Standarddatentyp:	numerisch, Zeichenketten
Strukturierter Datentyp:	Array, Record
abstrakter Datentyp:	Liste, Stack, Queue

Sie sind in Wertebereiche eingeteilt. Der Wertebereich ist abhängig von der verwendeten Plattform. z.B.: Datentyp Integer (Ganzzahl) bei

16 Bit-Plattform -32768 bis 32767,

32 Bit-Plattform -2.147.483.648 bis 2.147.483.647

## Datenstruktur

Objekt zur Speicherung und Organisation von Daten. Beschreibt wie die Daten intern angeordnet und verwaltet werden.

## Anweisungen

### Deklarieren

Beim Deklarieren werden die Dimension, der Bezeichner und der Datentyp der Variable festgelegt. Das muss vor der ersten Verwendung geschehen. Dabei wird der Speicher reserviert.

### Initialisieren

Nach dem deklarieren existiert die Variable, besitzt aber keinen definierten Wert. Der Variable wird ein Anfangswert zugewiesen.

In vielen Programmiersprachen haben Variablen nach dem Deklarieren den Wert NULL. **Aber nicht in jeder.** Es wird empfohlen einen definierten Anfangswert zu setzen. Andernfalls könnte die Variable einen zufälligen Wert besitzen!!!

## Kontrollstrukturen

Kontrollstrukturen dienen zum wiederholten Aufruf von Befehlen. In der Informatik kennen wir drei verschiedene Arten von Schleifen.

### Kopfgesteuerte Schleife

Die kopfgesteuerte Schleife ist üblicherweise eine WHILE-Struktur. Die Abbruchbedingung oder Laufbedingung wird vor dem Eintritt in die Schleife geprüft.

### Fußgesteuerte Schleife

Die fußgesteuerte Schleife ist eine DO- WHILE-Struktur. Die Abbruchbedingung ist am Ende des Konstruktes. Diese Schleife wird mindestens einmal durchlaufen.

### Zählschleife

In der FOR-Schleife wird festgelegt wie oft der Code im Konstrukt ausgeführt wird. Der Zähler wird bei jedem Durchlauf um eins erhöht, bis die Abbruchbedingung falsch ist.

## Bedingte Anweisung und Verzweigung

### IF-ELSE

IF-ELSE dient um die weitere Programmabarbeitung von einer Erfüllung oder Nichterfüllung eines logischen Ausdrucks abhängig zu machen. Ist der Ausdruck WAHR dann werden die Anweisungen hinter (THEN) ausgeführt, andernfalls die Anweisungen nach ELSE.

### SWITCH-CASE

Wird verwendet wenn bei einer Entscheidung mehrere Verzweigungen möglich sind. Der Ausdruck wird ausgewertet und mit Konstanten verglichen. Bei Gleichheit werden die Anweisungen der Konstante ausgeführt.



## Syntaktische Begriffe

### Programmiersprache

Ist eine Sprache zur Formulierung von Datenstrukturen und Algorithmen. Die Sprache setzt sich aus Anweisungen zusammen. (Syntax) Sehr oft werden die Anweisungen mittels Texteditor verfasst. Den erzeugten Text nennt man Quellcode oder Quelltext.

### Algorithmus

Er ist eine Verarbeitungsvorschrift die von einem mechanischen oder elektrisch arbeiteten Gerät abgearbeitet wird. Er besteht aus einer Reihe von Regeln für ein endliches Verfahren. Als Ergebnis soll eine spezifische Eingabe eine spezifische Ausgabe darstellen.

Als Eigenschaften muss er Eindeutigkeit, Ausführbarkeit, Endlichkeit, Determiniertheit und eine Terminierung besitzen. Beschreibung mittels Pseudocode, Struktogramme oder Flussdiagramm.

- ist eine Rechenvorschrift
- schrittweises Verfahren zur Lösung eines Problems

Ein **Algorithmus** besteht aus einer Folge von Befehlen, die der Reihe nach von der CPU abgearbeitet werden.

Ein Algorithmus kann wie eine Bedienungsanleitung sein. Versuchen Sie einen Algorithmus für nachstehendes Problem zu entwerfen.

### Auto starten und wegfahren

### Syntax

Ist die Art wie die Schlüsselwörter der Programmiersprache geschrieben werden.

Bsp.: Syntax Programmiersprache C :

int – Ganzzahl, printf ("Test") – Ausgabe, scanf() – Daten einlesen, /\* \*/ – Kommentar

### Programm

Anreihung von Befehlen in einer formalen Sprache die den Algorithmus von einer Eingabe in eine Ausgabe überführen. Ist eine Verfahrensvorschrift die meistens in binärer, ausführbarer Form im Dateisystem bereit steht.

Ein Programm übergibt dem Algorithmus Daten aus der Eingabe und gibt das Ergebnis aus.

### Prozess

Vorgang einer algorithmisch ablaufenden Informationsverarbeitung. Es beinhaltet Daten und Programme.

Unterschied Programm – Prozess:

Prozess      →      Flugzeug bauen

Programm → Plan zum bauen

Ein Prozess führt die Aufgabe aus. Das Programm ist das Werkzeug dazu.

## Vom Quelltext zum ausführbaren Programm

Damit ein Computer die Befehle aus dem Quelltext ausführen kann, muss dieser in eine Maschinensprache übersetzt werden. Das übersetzt übernimmt der Compiler oder ein Interpreter.

### Interpreter

Ein Interpreter liest den Quellcode ein, analysiert diesen, führt die Befehle aus und zeigt das Ergebnis an. Das alles passiert zur Laufzeit. Zu der Gruppe von Interpreter gehören Emulatoren und Kommandozeilen-Interpreter. Kommandozeilen-Interpreter unterstützen betriebssysteminterne Funktionen. Als Beispiel gibt es unter Windows die Eingabeaufforderung (cmd) und unter Linux und UNIX das Terminal.

### Compiler

Compiler übersetzen den Quellcode bevor das Programm ausgeführt wird. Der übersetzte Code ist noch nicht Lauffähig und wird dem Linker übergeben. Dieser bindet Standardfunktionen, die in bereitgestellten Bibliotheken angeboten werden ein. In Abbildung 1 ist ein einfacher Ablauf dargestellt.

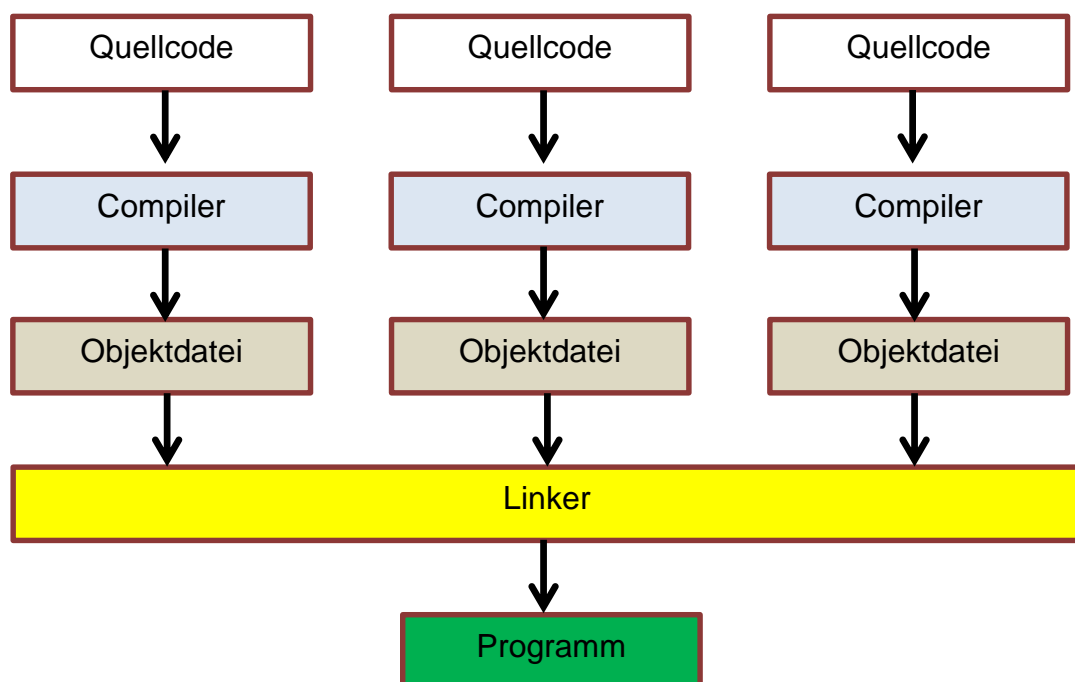


Abbildung 1: Einfacher Ablauf -- Quellcode - Programm

Laden Sie das Ergebnis (Quellcode) auf der Schulplattform Moodle hoch.

## Übung 1

In dieser Übung werden wir unter Linux ein einfaches C-Programm für die Ein- und Ausgabe erstellen

- Starten Sie die bereitgestellte VM
  - Benutzername: schueler Passwort:lbsvier
- erstellen Sie mittels Texteditor „NANO“ eine neue Datei
  - nano test.c (test.c ist der gewählte Dateiname)
- schreiben Sie ein Programm, dass ihren Namen auf der Konsole anzeigt
- Kompilieren Sie die Quelldatei mittels vorinstalliertem GCC
  - gcc *Quelldatei* -o *Name der ausführbaren Datei*
  - z.B.: gcc test.c -o hello.exe

## Übung 2

Schreiben Sie ein Programm, dass die Fläche eines Rechteckes berechnet.

Im ersten Schritt sollen die Werte den Variablen zugeteilt werden.

Im zweiten Schritt sollen die Länge und die Breite abgefragt werden und danach das Ergebnis ausgegeben werden.

Die Ausgabe sollte so wie in Abbildung 2 sein.

```
C:\Users\Herbert\Downloads\pocketgcc>
C:\Users\Herbert\Downloads\pocketgcc>
C:\Users\Herbert\Downloads\pocketgcc>
C:\Users\Herbert\Downloads\pocketgcc>laenge.exe

*** Flaechenberechnung ***
*** Laenge: 12
*** Breite: 23
*** Flaechen= 276
C:\Users\Herbert\Downloads\pocketgcc>
```

Abbildung 2: Fläche berechnen

### Anmerkung:

**Printf ()** dient zur Ausgabe von Zeichenketten, Zahlen oder Variablen. Wenn Variablen auf der Konsole ausgegeben werden soll, dann muss der Funktion der Datentyp bekanntgemacht werden.

Beispiel:

`int summe = 12; //Variable Summe besitzt den Wert 12 und ist ein Integer`

`printf("%i", summe);` //%i verweist auf einen Integer an, darauf folgt ein Komma und der Name der Variable

**Scanf()** dient zur Eingabe von Zeichenketten von der Tastatur. In dieser Funktion muss ebenfalls der erwartete Datentyp angegeben werden.

Beispiel:

*int laenge = 0; //Variable Deklarieren und initialisieren.*

*Scanf(„%i“, &laenge);//* %i ist Platzhalter für einen Integer.

Nach dem Komma den Adressoperator(&) und danach der Name der Variable.

Formatzeichen für Integer: %i, %d

Formatzeichen für Zeichenkette: %s

Formatzeichen für ein einzelnes Zeichen: %c

Formatzeichen für double und float: %f