

# Virtuelle Lernumgebung - Übung 2

## Aufgabe

Wählen Sie ein beliebiges Fachgebiet aus einem beliebigen Fach der gymnasialen Oberstufe und untersuchen Sie dieses unter dem Aspekt der vor-universitären Voraussetzungen genauer.

- a) Geben Sie für ein ausgewähltes Teilgebiet die erwarteten Kenntnisse im Sinne der Stufung kognitiver Ziele an.
- b) Ordnen Sie notwendige Fachbegriffe zu und erzeugen Sie daraus ein Begriffsnetz
- c) Erstellen Sie 3 bis 5 Fragen/Aufgaben verschiedenen Typs, die Ihrer Meinung nach zur Überprüfung der o. g. Ziele geeignet sind.

## Lösung

Fachoberschule

Lernbereich 2: Algorithmen und Programme

Lernbereich 2: Algorithmen und Programme	20 Ustd.
<p>Kennen von Programmiersprachen als Schnittstelle zwischen Mensch und Maschine</p> <ul style="list-style-type: none"><li>- Grundbegriffe</li><li>- historische Entwicklung von Programmiersprachen</li><li>- Klassifizierung von Programmiersprachen</li></ul> <p>Anwenden ausgewählter Problemlösungsstrategien in einer höheren Programmiersprache</p> <ul style="list-style-type: none"><li>- Visualisierung von Programmstrukturen</li><li>- komplexe Algorithmen</li><li>- einfache und strukturierte Datentypen</li><li>- Modularisierung</li></ul>	<p>Syntax, Semantik Compiler, Interpreter</p> <p>imperative Sprachen, deklarative Sprachen</p> <p>höhere Programmiersprachen: C, C++, C#, Delphi, Java, Javascript, Lazarus, PHP, PASCAL</p> <p>⇒ Problemlösestrategien</p> <p>→ OS INF, KI. 8, LB 2 Struktogramm, Ablaufplan, <b>GRAFCET</b> Folge, Verzweigung, Wiederholung</p> <p>euklidischer Algorithmus, Sortieralgorithmen</p> <p>Typmerkmale</p> <p>Unterprogrammtechnik</p> <p>⇒ Arbeitsorganisation: Arbeit im Team</p>

# Kognitive Lernziele

## Anforderungsbereich 1

Kennen von Programmiersprachen als Schnittstelle zwischen Mensch und Maschine.

Unterscheiden der Begriffe Syntax und Semantik sowie Compiler und Interpreter.

Unterscheiden von imperativen und deklarativen Programmiersprachen.

SuS charakterisieren einfache und strukturierte Datentypen anhand von typischen Merkmalen.

## Anforderungsbereich 2

Anwenden ausgewählter Problemlösungsstrategien in einer höheren Programmiersprache.

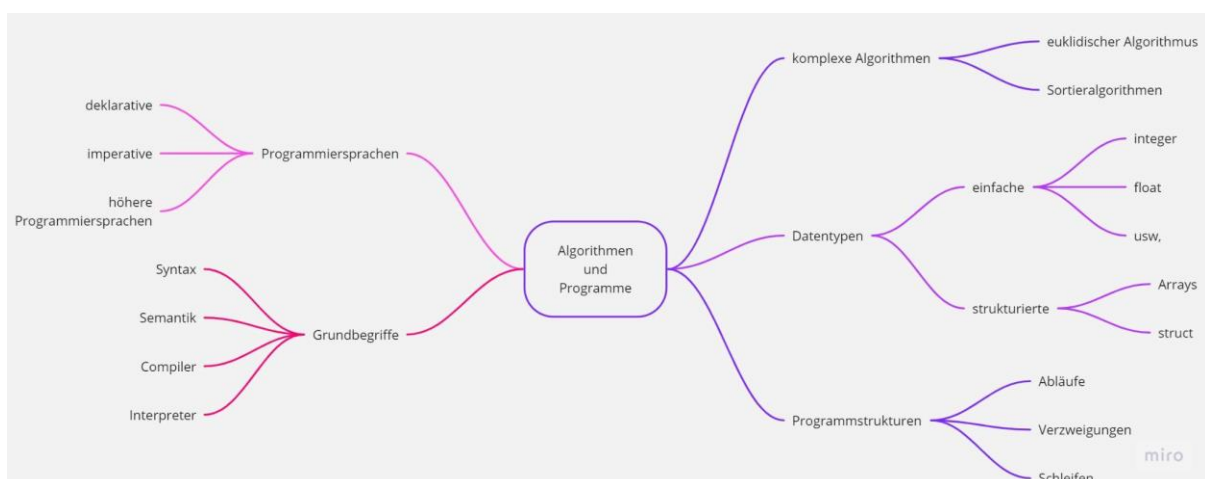
SuS visualisieren diese Problemlösungen anhand von Ablaufplänen oder Pseudocodes.

Die Schüler setzen algorithmische und heuristische Verfahrensweisen der Problemlösung für komplexe fachrichtungsspezifische Aufgabenstellungen ein (z. B. euklidischer Algorithmus oder Sortieralgorithmen) und setzen diese in Programmierungsumgebungen oder Pseudocodes um.

## Anforderungsbereich 3

SuS bewerten Problemlösungen unter verschiedenen Aspekten kritisch.

SuS entwickeln Problemlösungen in Programmstrukturen mit algorithmischen Werkzeugen wie Abläufen, Folgen, Verzweigungen und Schleifen.



# Aufgaben:

## Anforderungsbereich 1:

Erklären Sie den Unterschied zwischen einem Compiler und einem Interpreter.

Nennen Sie mindestens fünf Programmiersprachen und ordnen Sie zu, ob diese kompiliert oder interpretiert oder beides werden.

Unterscheiden Sie einfache und strukturierte Datentypen anhand von mindestens zwei Merkmalen.

Unterscheiden Sie imperative und deklarative Programmiersprachen und nennen Sie jeweils zwei Beispiele.

## Anforderungsbereich 2:

Verbundaufgabe zum Algorithmus rechts. (Quelltext erklären)

- Ordnen Sie mindestens zwei Kontrollstrukturen korrekt zu.
- Erklären Sie in kurzen Sätzen, was folgender Algorithmus (rechts) tut.
- Geben Sie die Variablenbelegung nach der Eingabe  $a := 3$  nach Ablauf des Algorithmus an.

Geben Sie an und begründen Sie anhand dreier Merkmale, zu welchem Programmierparadigma folgender Quelltext gehört:

```
def arithmetischesMittel(a = 0, b = 0, c = 0):
```

```
    mittel=(a+b+c)/3  
    return mittel
```

```
def summe(a=0, b=0):  
    return a+b
```

```
a=float(input("Zahl 1: "))  
b=float(input("Zahl 2: "))  
c=float(input("Zahl 3: "))
```

```
print(summe(a, b,))  
print(arithmetischesMittel(a, b, c))
```

## Anforderungsbereich 3

Erstellen Sie einen Pseudocode oder PAP zum Bubble-Sort-Algorithmus.

