

Das sollten Sie können

Liste der sach- und methodenkompetenzorientierten Lernziele

von-Neumann-Architektur

- 1.1 Sie können einen von-Neumann-Rechner definieren.
- 1.2 Sie können einen Single-Purpose-Processor und einen General-Purpose-Processor vergleichen.
- 1.3 Sie können die Komponenten eines von-Neumann-Rechners benennen und deren Aufgaben wiedergeben.
- 1.4 Sie können den Speicheraufbau des von-Neumann-Rechners erläutern.

Hard- und Software

- 2.1a Sie können die Begriffe *Hardware*,
- 2.1b *Software*,
- 2.1c *Maschinensprache* und
- 2.1d *Programmiersprache* definieren.
- 2.2 Sie können Software nach ISO/IEC 2382 kategorisieren.
- 2.3 Sie können die Vorgänge beim Start eines Python-Programms beschreiben.

Algorithmen

- 3.1 Sie können den Begriff *Algorithmus* definieren.
- 3.2 Sie können vier Darstellungsformen von Algorithmen benennen.
- 3.3 Sie können vorgegebene Struktogramme und Programmablaufpläne interpretieren.
- 3.4 Sie können zu gegebenen Problemen selbst Struktogramme erstellen.

Programmierung

Variablen und Datentypen

- 4.1.1 Sie können den Begriff Variable definieren.
- 4.1.2 Sie können Eigenschaften einer Variable benennen und angeben, wozu man Variablen verwendet.
- 4.1.3 Sie können den Begriff Datentyp definieren.
- 4.1.4a Sie können die Begriffe *Integer / Ganzzahl / int*,
- 4.1.4b *Floating Point Number / Gleitkommazahl / float*,
- 4.1.4c *String / Zeichenkette / str* und
- 4.1.4d *boolescher Wert / bool* definieren und Beispiele angeben.
- 4.1.5 Sie können angeben, wie man in Python Variablen Werte zuweist und in Ausdrücken verwendet.
- 4.1.6 Sie können den Unterschied zwischen einer Gleichung und einer Wertzuweisung erklären.
- 4.1.7 Sie können Operatoren für Zahlentypen angeben und deren Funktion benennen.
- 4.1.8 Sie können angeben, wie in Python Typumwandlungen durchgeführt werden.
- 4.1.9 Sie können erklären, wie man in Python den input-Befehl verwendet.
- 4.1.10 Sie können die Wertbelegung von Variablen im Laufe eines Python-Programms nachvollziehen.
- 4.1.11 Sie können Variablen in ihren Programmen so einsetzen, dass ihr Programmcode gut lesbar ist.

Kommentare

- 4.2.1 Sie können den Verwendungszweck von Kommentaren und die Python-Syntax von Kommentaren angeben.
- 4.2.2 Sie können in eigenen Programmen Kommentare zielgerichtet einsetzen, dass sie die Verständlichkeit erhöhen.

Algorithmen

- 4.3.1 Sie können den Begriff *Ausdruck* im Informatik-Zusammenhang definieren.
- 4.3.2 Sie können ein Struktogramm in ein Python-Programm überführen und umgekehrt.
- 4.3.3 Sie können den Ablauf eines Python-Programms aus seinem Quelltext nachvollziehen und ggf. Ausgaben benennen

- 4.3.4 Sie können neben der Python-spezifischen Variante eine Befehlssequenz angeben, mit der die Werte zweier Variablen getauscht werden.

Zeichenketten

- 4.4.1 Sie können erläutern, wie man in Python die Länge einer Zeichenkette bestimmt.
- 4.4.2a Sie können erklären, wie man in einer Zeichenkette auf einen konkreten Buchstaben zugreift und
- 4.4.2b auf eine Sequenz zugreift.
- 4.4.3 Sie können zwei Zeichenketten konkatenieren (zusammenfügen).
- 4.4.4 Sie können die Funktion der Python-Befehle `ord()` und `chr()` benennen.

Zusatzbibliotheken und Mathematik

- 4.5.1 Sie können erklären, wie zusätzliche Bibliotheken („Module“) importiert werden können.
- 4.5.2 Sie können das Modul benennen, mittels dessen auf gängige mathematische Funktionen zugegriffen werden kann.

Verzweigungen

- 4.6.1a Sie können die Funktion der `if`-Anweisung benennen,
- 4.6.1b gegebene `if`-Anweisungen interpretieren und
- 4.6.1c `if`-Anweisungen zu gegebenen Problemen formulieren.
- 4.6.2 Sie können verschiedene boolesche Vergleichsoperatoren und deren Bedeutung in Form einer Wahrheitstabelle angeben.
- 4.6.3 Sie können die Operatorenrangfolge boolescher Operatoren in Python angeben.
- 4.6.4a Sie können erklären, wie in Python die Mehrfachverzweigung mit `elif` und `else` realisiert wird und
- 4.6.4b können selbst Mehrfachverzweigung implementieren.
- 4.6.5 Sie können Verzweigungen und Mehrfachverzweigungen im Struktogramm darstellen.

Schleifen

- 4.7.1 Sie können den Begriff *Schleifen* im Informatik-Zusammenhang definieren.
- 4.7.2 Sie können den Unterschied zwischen einer `for`- und einer `while`-Schleife erklären.
- 4.7.3 Sie können `while`- und `for`-Schleifen in einem Struktogramm darstellen.

Funktionen

- 4.8.1 Sie können die Begriffe *Funktion* und *Prozedur* definieren.
- 4.8.2 Sie können den Vorteil von *Funktionen* und *Prozeduren* beschreiben.
- 4.8.3 Sie können den Unterschied zwischen *Funktionen* und *Prozeduren* benennen, sowohl *Funktionen* als auch *Prozeduren* implementieren und aufrufen.

Listen und Arrays

- 4.9.1 Sie können den Begriff *Liste* definieren.
- 4.9.2 Sie können den Begriff *Array* definieren.
- 4.9.3 Sie können Gründe für die Verwendung von Listen angeben.
- 4.9.4 Sie können in Python eine Liste bzw. ein Array erstellen und neue Elemente in eine Liste bzw. Array einfügen.
- 4.9.5 Sie können die Datentypen Liste und String miteinander vergleichen
- 4.9.6 Sie können auf einzelne Elemente der Liste zugreifen.
- 4.9.7 Sie können ein zweidimensionales Array anlegen und auf einzelne Elemente zugreifen.