1.16 Rückblick auf Kapitel 1

Leitidee

Am Ende des Kapitels lohnt sich ein Blick zurück: Welche Fachbegriffe kennst du jetzt? Was kannst du schon sicher, was solltest du noch einmal üben? Die Tabellen auf den nächsten Seiten helfen dir dabei, zum Beispiel bei der Vorbereitung auf eine Klassenarbeit.

1.16.1 Die Basisbegriffe

Diese Begriffe nennt der Lehrplan als **Basisbegriffe**. Du solltest jeden davon mit eigenen Worten erklären und an einem kleinen Beispiel zeigen können.

Begriff	Bedeutung	Beispiel
Variable	benannter Speicherplatz für einen Wert	`alter = 15`
Operator	Zeichen für eine Rechen-, Vergleichs- oder Logikoperation	`+`, `%`, `==`, `and`
Eingabe	Daten, die das Programm von der Nutzerin oder dem Nutzer bekommt	`name = input("Name? ")`
Ausgabe	Daten, die das Programm anzeigt	`print("Hallo", name)`
Bedingung	Ausdruck, der wahr (`True`) oder falsch (`False`) ist	`alter >= 18`
Verzweigung	Je nach Bedingung werden unterschiedliche Anweisungen ausgeführt.	`if … elif … else`
for-Schleife	wiederholt Anweisungen für jeden Wert einer Folge (Zählschleife)	`for i in range(5):`
while-Schleife	wiederholt Anweisungen, solange eine Bedingung wahr ist	`while zahl > 0:`
Funktion	benannter Programmteil, der bei Aufruf ausgeführt wird	`def quadrat(x):`
Parameter	Platzhalter für den Wert, den man der Funktion beim Aufruf übergibt	`x` in `def quadrat(x):`
Rückgabewert	Ergebnis, das eine Funktion mit `return` zurückgibt	`return x * x`
Zeichenkette	Folge (Sequenz) von Zeichen	`"Hallo"`
Liste bzw. Feld	geordnete Zusammenfassung mehrerer Elemente zu einer Einheit	`[4, 7, 1]`
Index	Position eines Elements; die Zählung beginnt bei 0	`zahlen[0]`

Gymnasium Ottweiler

1.16.2 Kompetenz-Check

Kreuze ehrlich an, wie sicher du dich fühlst. Wo du „noch nicht“ ankreuzt, lies den passenden Abschnitt noch einmal und bearbeite seine Arbeitsaufträge.

Ich kann ...	sicher	teilweise	noch nicht
Variablen verwenden, um Daten unterschiedlicher Typen zu speichern.			
Ein- und Ausgabeanweisungen verwenden und Eingaben in Zahlen umwandeln.			
arithmetische, vergleichende und logische Operatoren verwenden.			
verknüpfte Bedingungen mit den Regeln von De Morgan verneinen.			
ein-, zwei- und mehrseitige Verzweigungen analysieren und programmieren.			
eine Mehrfachauswahl (`CASE OF`) mit `match/case` programmieren.			
for- und while-Schleifen analysieren, programmieren und passend auswählen.			
Funktionen mit Parametern und Rückgabewert verwenden und selbst schreiben.			
fremde Programme lesen, erklären, testen und erweitern.			
eigene Algorithmen entwerfen und als Programm umsetzen.			
Zeichenketten als Folge von Zeichen beschreiben und bearbeiten (Verkettung, Länge, Zugriff auf Zeichen).			
Listen anlegen, zwischen Index und Element unterscheiden und Elemente lesen und ändern.			
mit Schleifen über Zeichenketten und Listen laufen.			
Minimum, Maximum, Summe und lineare Suche auf Listen programmieren.			
im Team ein Programmierprojekt umsetzen.			
Berufe mit Bezug zur Softwareentwicklung recherchieren und vorstellen.			