

## DIGCOMP 3.4 Programmieren

**Planen und Entwickeln einer Folge von verständlichen Anweisungen für ein Computersystem, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen.**

### KNOWLEDGE

K1: weiß, dass Computerprogramme aus Anweisungen bestehen, die von Menschen nach strengen Regeln in einer Programmiersprache geschrieben werden.

K2: weiß, dass Programmiersprachen Strukturen bereitstellen, die es ermöglichen, Programmanweisungen nacheinander, wiederholt oder nur unter bestimmten Bedingungen auszuführen und sie zu gruppieren, um neue Anweisungen zu definieren

K3: weiß, dass Programme von Computergeräten/-systemen ausgeführt werden, die in der Lage sind, Anweisungen automatisch zu interpretieren und auszuführen

K4: weiß, dass Programme Ausgabedaten in Abhängigkeit von Eingabedaten erzeugen und dass unterschiedliche Eingaben in der Regel zu unterschiedlichen Ausgaben führen.

K5: Sind sich bewusst, dass Daten (z. B. Zahlen, Texte, Bilder, Töne), die von einem Programm verarbeitet werden sollen, zunächst richtig digitalisiert (d. h. digital kodiert) werden müssen.

K6: weiß, dass ein Programm, um seine Ausgabe zu produzieren, Daten im Computersystem, das es ausführt, speichert und verändert, und dass es sich manchmal unerwartet verhält (z. B. fehlerhaftes Verhalten, Fehlfunktion, Datenleck)

K7: weiß, dass der Bauplan eines Programms auf einem Algorithmus beruht, d. h. auf einer schrittweisen Methode, um aus einer Eingabe eine Ausgabe zu erzeugen.

K8: weiß, dass zur Entwicklung von Algorithmen verschiedene Problemlösungsstrategien verwendet werden.

K9: weiß, dass Algorithmen und folglich Programme dazu dienen, reale Probleme zu lösen; Eingabedaten modellieren die bekannten Informationen über das Problem, und Ausgabedaten liefern Informationen, die für die Lösung des Problems relevant sind. Es gibt verschiedene Algorithmen und folglich Programme, die das gleiche Problem lösen.

K10: weiß, dass jedes Programm Zeit und Platz (Hardware-Ressourcen) benötigt, um seine Ausgabe zu berechnen, abhängig von der Größe der Eingabe und der Komplexität des Problems.

K11: weiß, dass es Probleme gibt, die von keinem bekannten Algorithmus in angemessener Zeit exakt gelöst werden können; daher werden sie in der Praxis häufig mit Näherungslösungen behandelt (z. B. Anwendungen der Künstlichen Intelligenz).

### SKILLS

S1: Kann einer anderen Person genaue Anweisungen zum Sortieren eines Spielkartensatzes aufschreiben.

S2: In der Lage sein, eine Reihe von Programmblöcken zu kombinieren, um ein Problem zu lösen.

S3: Weiß, wie man Probleme in einer Abfolge von Anweisungen erkennt und Änderungen vornimmt, um sie zu beheben (z. B. um einen Fehler im Programm zu finden und zu korrigieren; Probleme mit der Ausführungszeit zu finden).

S4: In der Lage sein, Eingabe- und Ausgabedaten in einigen einfachen Programmen zu identifizieren

### ATTITUDIEN

A1: Bereitschaft zu akzeptieren, dass Algorithmen und damit Programme bei der Lösung der zugrunde liegenden Probleme möglicherweise nicht perfekt sind.

A2: Entwicklung des Bewusstseins, dass die Verantwortung für Softwarefehler bei den Softwareentwicklern liegen sollte

A3: Entwicklung eines Verständnisses dafür, wie man Handlungen und Bedingungen auch in natürlichsprachlichen Beschreibungen präzise und unmissverständlich ausdrücken kann