

Lernziele zur Programmierprüfung I

Die Programmierprüfung wird ähnlich wie die Übungen gestaltet sein. Aufgabenstellung und Abgabe erfolgen über Github. Es werden nur die bei den Übungen verwendeten Module einzusetzen sein (**math**, **cmath**, **numpy** und allenfalls **random**). Andere Module sind nicht zugelassen!

Während der Prüfung (ausserhalb Annahme/Abgabe der Übungen) ist kein Internet-Zugriff erlaubt. Unterstützung durch KI-Tools ist genauso wie der Austausch mit Menschen nicht erlaubt.

Erlaubt (und empfohlen) ist hingegen das vorgängige Klonen und aktualisieren (`git pull`) des `Python-2023-24-Repository`.

- 1) Einen Python-Editor, der die Funktionalität von **Spyder** bietet, beherrschen. Es soll Python 3 mit Python-Version ≥ 3.9 installiert sein
- 2) Über Github Classroom Aufgaben annehmen, herunterladen (`git clone`) und Lösungen abgeben (`git add`, `git commit`, `git push`).
- 3) Datentypen `int` (Ganzzahlen), `float` (Fliesskommazahlen), `complex` (komplexe Zahlen), `str` (Zeichenketten), `numpy.ndarray` (numpy-Arrays), `list` (Listen), `dict` (Wörterbücher) kennen und verwenden
- 4) Eingabe (`input`) und Ausgabe (`print`), sowie *f*-String Formatierung beherrschen
- 5) Die Funktionen `lower`, `upper`, `len` und `split` für Zeichenketten kennen
- 6) Eingaben in den erforderlichen Datentyp konvertieren.
- 7) Die `if`-Abfrage (inklusive `elif` und `else`) beherrschen
- 8) Den `range`-Befehl verwenden, um Bereiche zu erstellen (auch mit negativer Schrittweite)
- 9) Listen erzeugen und ergänzen (mittels `append`), sowie auf Listenelemente zugreifen
- 10) Das Schlüsselwort `in` für Listen und Zeichenketten verwenden
- 11) `for`- und `while`-Schleifen verwenden, um durch Bereiche und Listen zu iterieren, inklusive Schleifenabbruch mittels `break`
- 12) Wörterbücher erstellen und ergänzen (mittels `update`)
- 13) Die Liste aller Schlüssel bzw. aller Werte, bzw. aller Schlüssel/Werte-Paare für ein Wörterbuch erzeugen (mittels `keys`, `values` und `items` Eigenschaften)
- 14) Berechnungen mit Ganzzahlen, Fliesskommazahlen und komplexen Zahlen durchführen
- 15) Kenntnis der elementaren Rechenoperationen (inklusive modulo und Ganzzahldivision) und logischen Operatoren (`and`, `or`, `not`)
- 16) Kenntnis von mathematischen Funktionen wie `sqrt`, `cos`, `sin`, `ceil`, `floor`
- 17) Folgenglieder und Reihenglieder, sowie Grenzwerte von Reihen (näherungsweise) berechnen
- 18) Ganzzahl-Arithmetik einsetzen um Berechnungen einer Zahl mit beliebiger Genauigkeit durchzuführen

- 19) Laplace-Wahrscheinlichkeiten durch Zählen aller günstiger Fälle berechnen (Bsp. Dreifachwurf von Würfeln)
- 20) Mittels des `numpy`-Moduls mit Vektoren und Matrizen rechnen (inklusive Summen, Differenzen, Matrixmultiplikation, Matrixpotenz, Skalarprodukt, Matrixinverse, Matrixdeterminante, Eigenwerten und Eigenvektoren)
- 21) Listen und Zeichenketten auf einen Bereich einschränken, Streichmatrizen erstellen