

 <h2 style="text-align: center;">2. Schulaufgabe im Fach Programmierung</h2> <h3 style="text-align: center;">Teil 2: Programmieren</h3>			Punkte / 30
Name	Vorname	ET1 Klasse	27.02.2025 Datum

- Erlaubte Hilfsmittel: Schriftliche Unterlagen in analoger und digitaler Form, kein Internet
- Versehen Sie Ihre programmierten Skripte mit einem Kommentar mit Ihrem Namen.
- Die Bewertungseinheiten (BE) sind Richtwerte und können sich noch ändern.
- Bearbeitungszeit: 35 Minuten

Aufgabe 1: Tribonacci-Folge als Funktion programmieren

(8 BE)

Neben der bekannten mathematischen Fibonacci-Folge gibt es außerdem die **Tribonacci-Folge**. In dieser Folge ist jedes Glied die Summe der drei vorherigen Glieder.

Die ersten Glieder der Tribonacci-Folge lauten: 0, 1, 1, 2, 4, 7, 13, 24, ...

Die Tribonacci-Folge soll mithilfe einer Python-Funktion erzeugt werden. Die Funktion soll als **Übergabeparameter** die Länge der zu berechnenden Folge erhalten. Als **Rückgabewert** soll die Funktion eine Liste mit den berechneten Werten der Folge zurückliefern.

- Schreiben Sie ein Python-Skript `Aufgabe_1_Tribonacci.py` und definieren Sie eine Funktion mit einem sinnvollen Namen, welche das beschriebene Verhalten für Tribonacci-Folgen mit beliebiger Länge umsetzt.
- Rufen Sie die Funktion im Skript beispielhaft für eine Länge von 10 auf und überprüfen Sie den Rückgabewert der Funktion auf folgende Werte:

[0, 1, 1, 2, 4, 7, 13, 24, 44, 81]

Aufgabe 2: Programm zur Implementierung eines Telefonbuchs

(22 BE)

Sie sollen ein Programm entwickeln, das ein Telefonbuch realisiert. Hierfür soll das Telefonbuch als **Klasse** in einem Python-Programm mit dem folgenden **Attribut** definiert werden:

- **telefonnummern** (Dictionary): Die Namen im Telefonbuch als Schlüssel (String) und den zugehörigen Telefonnummern als Wert (String).

a) Definition der Klasse

- Schreiben Sie ein Python-Skript `Aufgabe_2_Telefonbuch.py` und definieren Sie eine Klasse `Telefonbuch`, welche das beschriebene Attribut in einer Konstruktormethode definiert. (3P)

b) Im Folgenden soll die Klassendefinition um folgende **Methoden** erweitert werden.

- **eintrag_hinzufügen(self, name, telefonnummer):**
Fügt einen Telefonbucheintrag hinzu.
- **telefonnummer(self, name):**
Liefert die entsprechende Telefonnummer zum Namen als Rückgabewert.
- **telefonbuch_speichern(self):**
Speichert das Telefonbuch zeilenweise in einer Textdatei.

- Erweitern Sie die Klassendefinition aus Teilaufgabe a) um die Definition der Methode **eintrag_hinzufügen(self, name, telefonnummer)**. Die Methode soll das Dictionary um einen Eintrag ergänzen. (3P)
- Erweitern Sie die Klassendefinition aus Teilaufgabe a) um die Definition der Methode **telefonnummer(self, name)**. Die Methode soll zum angegebenen Namen die entsprechende Telefonnummer als Rückgabewert zurückliefern. Falls der Name nicht im Telefonbuch enthalten ist, soll stattdessen „Kein Eintrag gefunden“ als Zeichenkette zurückgegeben werden. (4P)
- Erweitern Sie die Klassendefinition aus Teilaufgabe a) um die Definition der Methode **telefonbuch_speichern(self)**. Die Methode soll die Einträge des Telefonbuchs zeilenweise wie folgt in einer Textdatei speichern: (5P)

Michael Meier: 01234567890

Peter Maffay: 01234567890

...

- Erstellen Sie unterhalb der Klassendefinition ein Objekt der Klasse **TeLe fonbuch** und fügen Sie drei Einträge hinzu. (4P)
- Rufen Sie die Methode **telefonnummer()** für den zweiten Eintrag auf und geben Sie die Telefonnummer auf der Konsole aus. (2P)
- Rufen Sie die Methode **telefonbuch_speichern()** auf. (1P)

Viel Erfolg! 😊