

# Programmiervorkurs Aufgaben

xel@

WiSe 24/25

## Inhaltsverzeichnis

<b>1 Überblick</b>	<b>1</b>
<b>2 Aufgaben</b>	<b>1</b>
2.1 Fibonacci Zahlen . . . . .	1
2.1.1 Implementierung der Berechnung . . . . .	2
2.1.2 Erweitern der Berechnung . . . . .	2
2.2 Taschenrechner . . . . .	2
2.2.1 Operationen als Funktionen . . . . .	2
2.2.2 Main und Input . . . . .	2
2.2.3 Nullpunkte bestimmen . . . . .	2
2.2.4 Weitere mögliche Erweiterungen . . . . .	2
2.3 Hangman . . . . .	2
2.3.1 Computer vs Human . . . . .	2
2.3.2 Human vs Human . . . . .	3
2.3.3 Computer vs Human vs Human . . . . .	3
2.3.4 Weitere mögliche Erweiterungen . . . . .	3
<b>3 Was jetzt?</b>	<b>3</b>

## 1 Überblick

Nachdem du jetzt das Vorkursskript komplett durchgearbeitet hast, kennst du schon ziemlich viele Konzepte.

Hier findest du jetzt eine Sammlung an Aufgaben die so (zum Teil) auch später in IPI (Einführung in die praktische Informatik) oder IPK (Programmierkurs) gestellt werden können.

Die folgenden Aufgaben kannst du in beliebiger Reihenfolge machen, suche dir einfach was aus, das interessant für dich klingt.

**Wichtig:** Außer es ist explizit vorgegeben erstelle Funktionen, Variablen, ... so wie du sie für sinnvoll erachtest.

## 2 Aufgaben

### 2.1 Fibonacci Zahlen

Fibonacci ist eine Zahlenfolge, die euch in den ersten Semestern verfolgen wird. Allgemein funktioniert sie wie folgt:

$$\begin{aligned}F_0 &= 0 \\F_1 &= 1 \\F_n &= F_{n-1} + F_{n-2}\end{aligned}$$

Also wäre zum Beispiel  $F_2 = F_1 + F_0$ , also  $F_2 = 1 + 0 = 1$ . Oder weiter:

$$F = [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \dots]$$

Deine Aufgabe? Schreibe einen Fibonaccirechner.

### 2.1.1 Implementierung der Berechnung

Schreibe eine Funktion

```
int fibonacci (int number)
```

Sie sollte  $F_n$  berechnen und  $n$  aus der Kommandozeile einlesen.

### 2.1.2 Erweitern der Berechnung

Erweitere deinen Code so, dass du in der Kommandozeile ein  $m$  und ein  $n$  einliest und dann alle Fibonaccizahlen von  $F_m$  bis  $F_n$  ausgibt.

## 2.2 Taschenrechner

In Lektion 7 hast du dich mit Arithmetik beschäftigt. Jetzt bauen wir einen Taschenrechner, der sowohl die 'Standardfunktionen', also Addition, Subtraktion, Multiplikation und Division für dich durchführt, aber auch fortgeschrittenere Funktionen wie die Mitternachtsformel zum Nullpunktebestimmen beherrscht.

### 2.2.1 Operationen als Funktionen

Baue vier Funktionen, die die Standardoperationen durchführt:

```
float add (float number, float number)
```

```
float sub (float number, float number)
```

```
float mult (float number, float number)
```

```
float div (float number, float number)
```

Danach soll das Ergebnis ausgegeben.

### 2.2.2 Main und Input

Schreibe deine main Funktion, die in Abfragen erst die Operation und dann die jeweils relevanten Zahlen als Input nimmt.

Danach sollte die gewählte Operation ausgeführt werden.

### 2.2.3 Nullpunkte bestimmen

Schreibe eine Funktion, die Nullpunkte einer quadratischen Funktion bestimmt, entweder mit der pq- oder der abc-Formel.

Passende deine Main so an, dass Nullpunkte bestimmen auch eine Option mit passenden Inputs ist.

### 2.2.4 Weitere mögliche Erweiterungen

- Wurzelrechnen
- Matrizen addieren
- Matrizen multiplizieren
- ...

## 2.3 Hangman

Bastel dir ein Hangman Spiel zusammen. Einige mögliche Spielversionen wären:

### 2.3.1 Computer vs Human

Der Computer wählt aus einer Liste zufällig ein Wort und dann spielt man Hangman gegen den Computer.

### 2.3.2 Human vs Human

Das Spiel kriegt von Spieler 1 ein Wort als Input, Spieler 2 versucht dann das Wort zu erraten.

### 2.3.3 Computer vs Human vs Human

Der Computer wählt ein zufälliges Wort, zwei Spieler versuchen abwechselnd das Wort zu erraten, die Person, die das Wort errät gewinnt (oder der Computer, sollte Hangman zuerst gezeichnet worden sein).

### 2.3.4 Weitere mögliche Erweiterungen

- Es gibt mehrere Modi, am Anfang wird der Modus ausgewählt
- Eine ASCII Zeichnung, die bei einem falschen Buchstaben erweitert wird
- Eine Anzeige für schon eingegebene Buchstaben
- ...

## 3 Was jetzt?

Jetzt solltest du ein ziemlich gutes Verständnis für die grundlegende Programmierung haben!

Hast du eine eigene Idee, was du gerne programmieren würdest?

Dann versuche dich sehr gerne daran!

Ansonsten findest du bei Project Euler viele Mathe- und Programmier-Challenges, die du lösen kannst.

