

Arrays Üben 2

Aufgabe 1. Beispielarray erstellen und auswerten.

Für Demonstrationszwecke benötigen wir ein Array von Zahlen, die Temperaturen einer Wetterstation als Integer ausgibt.

- a) Recherchieren Sie einen sinnvollen Wertebereich für Temperaturen in Weimar.
- b) Es ist wenig glaubwürdig, wenn die Werte von -1 °C auf 28 °C steigen. Überlegen Sie sich einen Algorithmus, der glaubwürdige Änderungen der Werte realisiert.
- c) Schreiben Sie ein Programm, das fragt, wie viele Werte das Feld haben soll und dann Ihre Lösung aus b) umsetzt, um ein Feld mit Beispielwerten zu erzeugen. Geben Sie das Feld aus. Überlegen Sie sich Testfälle.
- d) Schreiben Sie nun ein Programm, das
 - a. Den Maximal- und Minimalwert findet.
 - b. Die größte und die kleinste Änderung der Werte. (Hinweis: Recherchieren Sie zum `abs()`-Befehl in Python).
 - c. Glättung durch gleitenden Durchschnitt: Die einfachste Art, Trends darzustellen ist der gleitende Mittelwert dabei wird der Wert an der Stelle i durch den Durchschnitt der Elemente an den Stellen $i-4, i-3, i-2, i-1$ und i ersetzt. Der Wert kann zum ersten Mal für $i = 4$ errechnet werden. Schreiben Sie ein Programm, das den gleitenden Mittelwert für ein Array berechnet.

Hinweis:

d-b: Berechnen Sie den Betrag der Differenz zwischen aktuellem Element und nachfolgendem Element. Das Array für den Mittelwert kann an den ersten vier Stellen die Werte -200 haben.

Aufgabe 2. Pokerblatt.

Die Blätter eines Pokerspiels haben vier Farben:

- Clubs (Kreuz)
- Diamonds (Karo)
- Spades (Pik)
- Hearts (Herz)

Jede Farbe hat folgende Karten: **2, 3, 4, 5, 6, 7, 8, 9, 10, Jack (Bube), Queen (Dame), King (König), Ace (Ass)**. Um ein Pokerspiel zu programmieren benötigt man einen Handchecker. Das ist ein Programm, das entscheidet, wessen Hand gewinnt. Um den Handchecker die Karten effektiv zu übermitteln wird zuerst die Karte und dann die Farbe zu einem String zusammengefügt, z.B. JC für Kreuz-Bube, 10D für Karo-10, QH für Herz-Dame und AS für Ace of Spades: Pik-Ass (Lemmy forever!)

- a) Ausgehend von den Arrays

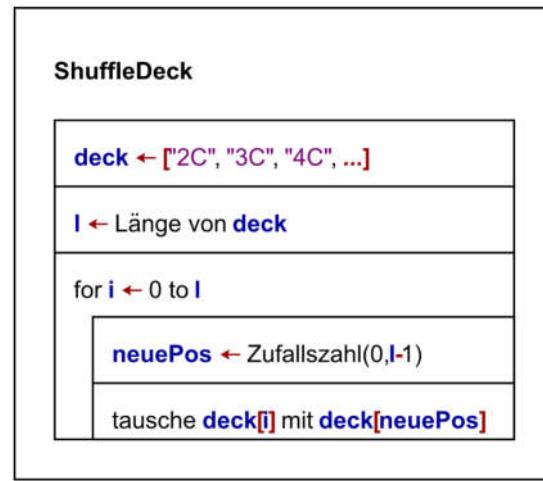
```
farben = ["C", "D", "S", "H"]
karten = ["2", "3", "4", "5", "6", "7", "8", "9", "10", "J", "Q", "K", "A"]
```

schreiben Sie ein Python-Programm, dass das Array deck erzeugt, das alle möglichen Kartencodes enthält:

```
deck = ["2C", "3C", "4C", ..., "QC", "KC", "AC", "2D", "3D", ..., "KD", "AD"]
```

Hinweis: Schreiben Sie eine Schleife in einer Schleife, die äußere Schleife für die Farben, die innere Schleife für die Karten.

- b) Jetzt müssen die Karten nur noch gemischt werden. Ein einfacher Algorithmus ist in dem Struktogramm abgebildet. Implementieren Sie den Algorithmus in Python.



Aufgabe 3. Bingo-Spielbrett.

Ein Bingo-Spielfeld besteht aus 5x5 Feldern, auf denen Zahlen von 1 bis 75 zufällig angeordnet sind. Die Abbildung zeigt, wie man die das Raster in einem einfachen Array abbilden kann.

Schreiben Sie eine Funktion `inalready(feld, wert)`, die prüft, ob ein Wert schon in einem Feld vorhanden ist. Ist dies der Fall, soll 1 zurückgegeben werden, ansonsten 0.

Hinweis: Tun sie sich selbst einen Gefallen und nutzen Sie keine eingebauten Funktionen, sondern versuchen Sie, selbst einen Algorithmus zu entwickeln.

Schreiben Sie ein Computerprogramm, das ein Array `spielfeld` mit 25 Feldern erzeugt, das mit Zahlen von 1 bis 75 gefüllt ist. Keine Zahl darf auf dem Spielfeld doppelt vorkommen, verwenden Sie dafür die selbstgeschriebene Funktion `inalready`.

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

Aufgabe 4. Vier Arrays.

Schreiben Sie ein Programm, dass vier Arrays mit folgenden Eigenschaften generiert:

- `feld1` enthält nur gerade ganze Zahlen,
- `feld2` enthält ganze Zahlen, die aufsteigend sortiert sind (Dopplungen sind möglich)
- `feld3` enthält ganze Zahlen, die absteigend sortiert sind (Dopplungen sind hier nicht möglich)
- `feld4` enthält nur ganze Zahlen, die durch 3 teilbar sind, und ist aufsteigend sortiert, wobei eine Dopplung nicht vorkommt.

Verwenden Sie keinen Sortieralgorithmus.