Blatt 6

Vorkurs Bauingenieurwesen - Informatik - 17.05.24

Heute spielen wir Sudoku.

Aufgabe 1

Das Spielfeld wird als Liste von Listen von `int` gespeichert. Die einzelne Zahl auf dem Spielfeld wird als `int` gespeichert, ein leerer Eintrag kann mit der Zahl `0` dargestellt werden. Jede Zeile von Einträgen ist als Liste gespeichert. Alle Zeilen sind dann erneut in einer Liste zusammengefasst.
Schreibe die Funktion `baue_spielfeld`, die ein Spielfeld erzeugt und zurückgibt.

```
def baue_spielfeld() -> list[list[int]]:
```

Das erzeugte Feld soll bereits diese Einträge haben:

```
    1 2 3   4 5 6   7 8 9
  +-------+-------+-------+
1 | 4 1   |   6 5 |     7 |
2 |     6 |     7 | 4 8   |
3 | 2   7 | 4 9   |     6 |
  +-------+-------+-------+
4 |   6   |   7   | 1     |
5 | 3   1 | 5     |   7 2 |
6 |   9   |   4 2 | 3   8 |
  +-------+-------+-------+
7 | 1   8 | 6     |   2 9 |
8 |   2   |   1 8 | 6 4   |
9 | 6     | 3     |   1   |
  +-------+-------+-------+
```

Aufgabe 2

Schreibe eine Funktion `zeige_spielfeld`, die das aktuelle Spielfeld auf der Konsole ausgibt. Eine mögliche Formatierung ist in Aufgabe 1 gegeben.

```
def zeige_spielfeld(
    spielfeld: list[list[int]]):
```

Aufgabe 3

Schreibe eine Funktion `gueltig`, die ein Sudokufeld übergeben bekommt und testet ob jede Zeile, jede Spalte und jeder Block jede Zahl nur ein Mal enthalten. Vorsicht: Die `0` ist für die Spiellogik keine Zahl, sondern ein leeres Feld.

```
def gueltig(spielfeld: list[list[int]])
    -> bool:
```

Umgesetzt werden kann die Funktion in drei Teilen. Zuerst wird für jede Zeile geguckt, ob eine Zahl mehrfach vorkommt. Dann für jede Spalte, dann für jeden Block.
Jeder Teil kann dabei mit geschachtelten `for`-Schleifen implementiert werden. Als Beispiel hat der Test für die Gültigkeit der Zeilen drei geschachtelte Schleifen.

1. für jede Zeile des Spielfeldes:
2. für jede Zahl von 1 bis inklusive 9:
3. lege eine Variable an, die mit zählt, wie oft diese Zahl in der Zeile vorkommt. Initialisiere die Variable mit 0.
4. für jeden Eintrag der aktuellen Zeile:
5. Wenn der Eintrag identisch zur aktuellen Zahl ist, addiere 1 zu der Zählvariablen hinzu.
6. Wenn die aktuelle Zahl mehr als ein Mal gefunden wurde, breche die Ausführung der Funktion ab und gebe `False` als Ergebnis zurück.

Wenn alle drei Teile durchgelaufen sind ohne abzubrechen, gebe `True` von der Funktion zurück.

Aufgabe 4

Schreibe eine Funktion `naechster_zug`, die zuerst die Eingabe eines Zeilenindexes, dann die Eingabe eines Spaltenindexes, dann die Eingabe eines Wertes an dieser Position erwartet. Bevor der Wert eingegeben werden kann, soll geprüft werden, ob die angegebene Position überhaupt frei ist. Nachdem der Wert eingegeben wurde, soll geprüft werden, ob das Spielfeld noch gültig ist. Nur dann wird die Funktion `naechster_zug` beendet. Sonst muss der Eintrag wieder gelöscht werden und eine neue Eingabe erwartet werden.

```
def naechster_zug(
    spielfeld: list[list[int]]):
```

Aufgabe 5

Schreibe eine Funktion `spielen` ohne Argumente und ohne Rückgabewerte. Als lokale Variable soll ein Spielfeld erzeugt werden, dass dann in einer Schleife vom Spieler gefüllt wird. Die Funktion soll verlassen werden, wenn das Sudoku gelöst ist.

```
def spielen():
```