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_um`, die ein Sudokufeld, sowie eine Position im Sudokufeld (Zeile und Spalte) übergeben bekommt und testet ob der Wert dort sowohl in der Zeile, als auch in der Spalte, als im entsprechenden Block nur ein Mal enthalten ist.

```
def gueltig_um(spielfeld: list[list[int]],
    zeile_nr: int, spalte_nr: int) -> bool:
```

Umgesetzt werden kann die Funktion in drei Teilen. In Teil 1 muss geguckt werden ob wo anders in der selben Zeile noch Mal der selbe Wert steht. Wenn ja: `return False`. In Teil 2 passiert selbiges für die Spalte und in Teil 3 für den Block der die Position enthält. Wenn weder die Zeile, noch die Spalte, noch der Block den Wert mehrfach enthält, gebe `True` zurück.
Tipp: die Zeilen des Blockes beginnen an Index `zeile_nr - (zeile_nr % 3)` und die Spalten des Blockes an Index `spalte_nr - (spalte_nr % 3)`.

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 (um die geänderte Position). 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():
```