

Lehrziele:

- Schleifen
- Strings und Char

Aufgabenstellung

Schreibe ein Programm „Turtle.cs“ mit dem man eine kleine Schildkröte (das Zeichen ,*‘) am Bildschirm bewegen kann. Dazu gibt der Benutzer beim Programmstart einen String ein, der die Definition der Bewegung enthält. Jedes Zeichen des Strings enthält eine Anweisung (=Kommando), was die Schildkröte machen soll:

- Zahlen von 0 bis 7 bewegen die Schildkröte in eine Richtung

7	0	1
6	*	2
5	4	3

Mit ,3‘ wird die Schildkröte nach rechts unten bewegt.

- ,D‘ (für PenDown) schaltet das Zeichnen eines ,*‘ nach jedem Kommando ein.
- ,U‘ (für PenUp): Jedes weitere Kommando verändert nur die Position ohne ein ,*‘ auszugeben.

Beispiel 1:

Mit der Eingabe 1111D222 zeichnet das Programm eine waagrechte Linie. Mit 1111 geht man vom Ursprung 0.0 (links/unten) schräg nach oben (4 mal wegen 1111), mit D schaltet man das Zeichnen ein - dabei wird schon ein ,*‘ ausgegeben. 222 zeichnet dann weiteren drei ,*‘ der Linien:

```
* * * *
```

Beispiel 2:

Mit der Eingabe U1111D01234567 zeichnet das Programm einen Kreis:

```
* *
*   *
*   *
*   *
* *
```

Beispiel 2:

Diese Eingabe U1111D0011223344556677U2222222222D0011223344556677U66000D44444U566D222222 zeichnet ein Gesicht:

```

* * *           * * *
*   *   *   *   *
*       * * *   *
*       * * *   *
*       * * *   *
*   *   *   *   *
* * *   *   * * *
* * * * * *

```

Hinweise

- Beim Programmstart ist die Schildkröte auf Position 0:0 und es gilt ‚U‘ (penUp)
- Vor dem Zeichnen (vor dem ersten Kommando) muss die Konsole initialisiert werden. Verwende dazu folgende Anweisungen:

```
Console.SetWindowSize(80, 25);  
Console.Clear();
```

- Man kann die Position der Ausgabe mit der nächsten Anweisung festlegen:

```
Console.SetCursorPosition(x, y);  
Console.Write('*');
```

ACHTUNG: Der Ausgangspunkt 0:0 ist links/**unten**.

Beim Aufruf von `SetCursorPosition` wächst aber die Y Achse nach unten (0:0 ist links/**oben**). Man kann diese „Spiegeln“ indem man für Y „25-1 – y“ berechnet.

(25 ist die Maximalgröße, -1 weil man die Position ab 0 angibt und y im Wertebereich 0..24 sein darf)

- Am Ende des Programms sollte die folgende Anweisung stehen:

```
Console.SetCursorPosition(0, 24);  
Console.WriteLine();
```

Damit wird verhindert, dass das gerade Gezeichnete überschrieben wird.

Zusatzaufgaben

- Überprüfe die Positionen.
Ignoriere Anweisungen, die eine ungültige Position ergeben (x: 0..79, y: 0..24)
- Die Eingabe kann durch mehrere Zeilen erfolgen. Jede einzelne Eingabezeile wird zu einer Gesamteingabe verbunden. Die Eingabe ist beendet, wenn der Benutzer eine Leerzeile eingibt.
- Unterstütze die Ausgabe von Farben.
Es kann z.B. mit dem Eingabe-Zeichen R auf „rot“ umgeschaltet werden.
- Definiere verschiedene Standartausgaben.
Gibt der Benutzer z.B. \$1 ein, wird wie in Beispiel 1 ein Kreis gezeichnet.