Unspezifische Ereignisse

Funktionen höherer Ordnung

Dieses Jupyter Notebook demonstriert den Umgang mit Funktionen höherer Ordnung in dritten Programmkomponenten.

Die exemplarische dritte Programmkomponente ist hier ein Übersetzer von Ereignisse in MIDI Dateien.

Zuerst müssen die spezifischen Module geladen werden.

[2]:
```
from mutwo import core_events
from mutwo import midi_converters
from mutwo import music_parameters
```

Als nächstes wird für dieses Beispiel die Klasse **Note** definiert. Sie soll eine Note im Kontext westlicher Musik repräsentieren.

[3]:
```
class Note(core_events.SimpleEvent):
    # Dynamik ist konstant
    volume = music_parameters.WesternVolume('p')

    def __init__(self, pitch, duration):
        self.pitch = pitch

        # Das ist Python spezifischer Syntax, um dem Konstruktor
        # der Basisklasse das "duration" Argument zu übermitteln.
        super().__init__(duration)
```

Jetzt wird eine einfache Melodie aus zwei Noten definiert.

[4]:
```
melody = core_events.SequentialEvent(
    [
        Note(music_parameters.WesternPitch('c'), 1),
        Note(music_parameters.WesternPitch('d'), 1),
    ]
)
```

Jetzt soll die Melodie in eine MIDI Datei übersetzt werden.

In der API Dokumentation der Klasse *EventToMidiFile* kann nachgelesen werden, dass diese unter anderem mit dem Argument *simple_event_to_pitch_list* initialisiert wird. Wir können auch nachlesen, dass dessen Standardwert davon ausgeht, dass ein *SimpleEvent* ein Attribut namens *pitch_list* hat, was *Note* aber nicht kennt. Deswegen müssen wir das Argument überschreiben, sodass es unser *pitch* Attribut finden kann.

[5]:
```
# Definiere zuerst den Übersetzer
event_to_midi_file = midi_converters.EventToMidiFile(
    simple_event_to_pitch_list=lambda simple_event: [
        getattr(simple_event, 'pitch')
    ]
)
# Übersetze jetzt die Melodie
event_to_midi_file.convert(melody, 'my_melody.mid')
```