

## Exercise Sheet 4: Generate MIDI Files from Guitar Tabs

### Organizational issues

Keep in mind that the chosen **representative** of your group hands in your solution as Moodle does **not yet support group solutions**. Always make sure to include your names in a groupmembers.txt in your solution.

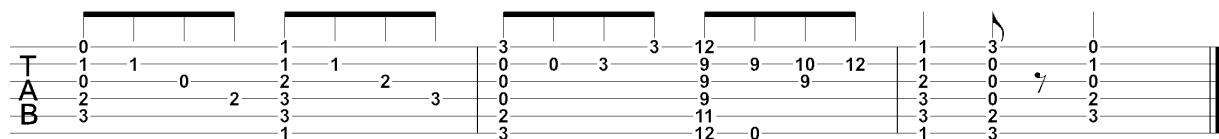
*Hint:*

*If you have questions/problems solving this task, use the reference manual available here:*

<http://www.se-rwth.de/publications/MontiCore-5-Language-Workbench-Edition-2017.pdf>  
*to find further explanations.*

### Background

Guitarists often use so-called "tabs" to exchange music<sup>1</sup>. Each line in the staff represents a string on the guitar. Each number indicates the fret in which the guitarist should play the string. Zero refers to playing an open string. Tabs can also contain length information by using staves.



Online, guitarists exchange tabs for their favorite songs mostly in ASCII notation. The tab shown here would look like this in ASCII notation:

```
e|-0-----1-----3-----3-12-----1---3--0-----|
B|-1-1---1-1---0-0-3---9-9-10-12---1---0--1-----|
G|-0--2--2--2---0-----9---9-----2---0--0-----|
D|-2---2-3---3--0-----9-----3---0--2-----|
A|-3-----3-----2-----11-----3---2--3-----|
E|-----1-----3-----12-0-----1---3-----|
```

<sup>1</sup> [https://en.wikipedia.org/wiki/Tablature#Guitar\\_tablature](https://en.wikipedia.org/wiki/Tablature#Guitar_tablature)

One problem with these tabs, however, is that it is difficult to detect errors in them. Since it is often easier to hear an error than to see it in the tab, the goal of this exercise sheet is to read in tabs and create a playable MIDI file from them.

Therefore, you will use MontiCore to generate Python code that generates a MIDI File.

### Exercise 4.1: Generate MIDI files (7 Points)

We already created a MontiCore grammar, that can be used to parse tabs in the notation used by ultimate-guitar.com<sup>2</sup>.

1. Use the class `MidiPythonGenerator` to generate a python file using MontiCore's `GeneratorEngine`
2. The Freemarker template `MidiGenerator.ftl` already contains some boilerplate code for generating MIDI files. Extend it to generate a MIDI file that corresponds to the given guitar tab.

Assumptions and additional information:

- We assume that all music has **4 beats per bar**.
- We assume that all **tabs only contain fret numbers** and no additional information such as "hammer-on" or "slide up".
- Assume that **each note has a duration of one beat**.
- Assume that the guitar has standard tuning (E-A-D-G-B-E).
- MIDI pitch 60 refers to the middle C<sup>3</sup>.
- Going one fret higher corresponds to increasing the pitch by one.
- Tab files can contain multiple staves that should be played sequentially.
- We use "MIDIUtil" for creating MIDI files. It's documentation and an example can be found online<sup>4</sup>. You can use `addNote()` for adding single notes to your MIDI.
- Notes that are on top of each other should be played at the same time.
- The generated Python code should work with **Python 3.5**
- `ASTSixStrings::getAllNotes()` provides an ordered list of all notes that should be played together

### Exercise 4.2: Generate notes of different lengths (3 Points)

In reality not all notes have the same length. Therefore, we extended the DSL used by ultimate guitar to include note durations.

<sup>2</sup> [https://www.ultimate-guitar.com/lessons/for\\_beginners/guitar\\_tabs\\_template.html](https://www.ultimate-guitar.com/lessons/for_beginners/guitar_tabs_template.html)

<sup>3</sup> <https://www.dummies.com/art-center/music/guitar/how-to-tune-a-guitar-from-the-piano/>

<sup>4</sup> <https://code.google.com/archive/p/midiutil/>

```
e|-0-----1-----3-----3-12-----1---3---0-----|
B|-1-1---1-1---0-0-3---9-9-10-12---1---0---1-----|
G|-0--2--2--2---0-----9--9-----2---0---0-----|
D|-2--2-3---3--0-----9-----3---0---2-----|
A|-3-----3-----2-----11-----3--2---3-----|
E|-----1-----3-----12-0-----1---3-----|
L|-8-888-8-888--8-8-8-8-8--8-8--8---4---8-8-2-----|
```

The additional “L”-Line denotes the length of the notes on top of it measured in “fraction of a bar”. In the above example, most notes have a length of  $\frac{1}{8}$  of a bar.

Extend your generator as follows:

- If a staff does not include the length line, each note should have a duration of one beat.
- If a staff includes the length line, each note should have the length given by the length line.

As a solution of this task, zip and hand in your project **without** the generated target, i.e., after running `mvn clean`. Make sure the generated files are reproducible by just running `mvn install`. **Not working projects, e.g., projects that have test failures or compile errors, will not gain you any points.**