# example06(MIDI Filer)

# プログラム概要

本プログラムはJavaFXとMIDIの練習のために作成しています。

YAMAHA MDFの機能を参考にexample01をもとに作成しています。

機能は以下の通りです。

・SysexMessageを受信しSMF形式でファイルに書き出す。

・SMF形式のファイルを読込みSysexMessageを送信する。

## YAMAHA MDFとは？

### MDF1

### MDF2

### MDF2

## SMFとは？

# 機能概要

## 画面構成

### メイン画面

### MIDI設定ダイアログ

### 出力ファイルダイアログ

### 入力ファイルダイアログ

## ファイル構成

### フォルダー構成

# プログラム詳細

## MIDIデバイス

MIDIデバイスはMidiSystem.getMidiDeviceInfo()でデバイス情報を取得することができる。

MidiDeviceがハードウェアMIDIポートを表しているかどうかを判断するには以下の方法を用いる。

MidiDevice device = ...;

if ( ! (device instanceof Sequencer) && ! (device instanceof Synthesizer)) {

// we're now sure that device represents a MIDI port

// ...

}

また入力デバイス／出力デバイスの判定はTransmitter、Receiverの有無で判定する。

それぞれのデバイス情報は以下のように取得する。

/\*\*

\* 入力デバイス情報リストの取得

\*

\* @return デバイス情報リスト

\*/

public static MidiDevice.Info[] getInputMidiDeviceInfo() {

ArrayList<MidiDevice.Info> list = new ArrayList<>();

MidiDevice.Info[] infos = MidiSystem.getMidiDeviceInfo();

for (int i = 0; i < infos.length; i++) {

// throws MidiUnavailableException

try {

MidiDevice device = MidiSystem.getMidiDevice(infos[i]);

// getMaxTransmitters()が0以上の場合は入力デバイス

// Synthesizer、Sequencer以外は

if (device.getMaxTransmitters() > 0 && !(device instanceof Synthesizer)

&& !(device instanceof Sequencer))

list.add(infos[i]);

} catch (MidiUnavailableException e) {

e.printStackTrace();

}

}

return list.toArray(new MidiDevice.Info[0]);

}

/\*\*

\* 出力デバイス情報リスト取得

\*

\* @return

\* @throws MidiUnavailableException

\*/

public static MidiDevice.Info[] getOutputMidiDeviceInfo() throws MidiUnavailableException {

ArrayList<MidiDevice.Info> list = new ArrayList<>();

MidiDevice.Info[] infos = MidiSystem.getMidiDeviceInfo();

for (int i = 0; i < infos.length; i++) {

// throws MidiUnavailableException

MidiDevice device = MidiSystem.getMidiDevice(infos[i]);

// getMaxReceivers()が0以上の場合は出力デバイス

// Synthesizer、Sequencer以外はハードウェアMIDIポート

if (device.getMaxReceivers() > 0 && !(device instanceof Synthesizer) && !(device instanceof Sequencer))

list.add(infos[i]);

}

return list.toArray(new MidiDevice.Info[0]);

}

## MIDI入力

MIDI入力はReceiverインターフェースから作成し、send()メソッドを実装する。

example01ではメッセージの表示のみ行っていたためにデータの保持が不要でしたが、ファイル保存のためにメッセージの保持が必要となるのでMidiEventでQueueを作成する。

また不要なメッセージを取得しないように、それぞれのメッセージ毎に有効／無効の設定しフィルタリングする。

public void send(MidiMessage message, long timeStamp) {

if (message != null) {

// 受信メッセージをフィルタリングします。

if (message instanceof ShortMessage) {

switch (message.getStatus()) {

case ShortMessage.NOTE\_OFF: // 0x80

if (!isNoteOffActive()) {

return;

}

break;

case ShortMessage.NOTE\_ON: // 0x90

if (!isNoteOnActive()) {

return;

}

break;

case ShortMessage.POLY\_PRESSURE: //0xA0

if (!isPolyPressureActive()) {

return;

}

break;

case ShortMessage.CONTROL\_CHANGE://0xB0

if (!isControlChangeActive()) {

return;

}

break;

case ShortMessage.PROGRAM\_CHANGE://0xC0

if (!isProgramChangeActive()) {

return;

}

break;

case ShortMessage.CHANNEL\_PRESSURE://0xD0

if (!isChannelPressureActive()) {

return;

}

break;

case ShortMessage.PITCH\_BEND://0xE0

if (!isPitchBendActive()) {

return;

}

break;

case ShortMessage.MIDI\_TIME\_CODE://0xF1

if (!isMidiTimeCodeActive()) {

return;

}

break;

case ShortMessage.SONG\_POSITION\_POINTER://0xF2

if (!isSongPositionPointerActive()) {

return;

}

break;

case ShortMessage.SONG\_SELECT://0xF3

if (!isSongSelectActive()) {

return;

}

break;

case ShortMessage.TUNE\_REQUEST://0xF6

if (!this.isTuneRequestActive()) {

return;

}

case ShortMessage.END\_OF\_EXCLUSIVE://0xF7

if (!this.isEndOfExclusiveActive()) {

return;

}

break;

case ShortMessage.TIMING\_CLOCK://0xF8

if (!isTimingClockActive()) {

return;

}

break;

case ShortMessage.START://0xFA

if (!isStartActive()) {

return;

}

break;

case ShortMessage.CONTINUE://0xFB

if (!isContinueActive()) {

return;

}

break;

case ShortMessage.STOP://0xFC

if (!isStopActive()) {

return;

}

break;

case ShortMessage.ACTIVE\_SENSING://0xFE

if (!isActiveSensingActive()) {

return;

}

break;

case ShortMessage.SYSTEM\_RESET://0xFF

if (!this.isSystemResetActive()) {

return;

}

break;

}

} else if (message instanceof SysexMessage) {

if (!isSysexMessageActive()) {

return;

}

}

inputMidiEventQueue.offer(message);

}

}

## MIDI出力

## SMF

# 参照文献

・YAMAHA MDF1ユーザマニュアル

・MIDI1.0規格書 - 音楽電子事業協会

http://amei.or.jp/midistandardcommittee/MIDIspcj.html