```
require 'java'

java_import 'javax.swing.JFrame'
java_import 'javax.swing.JMenu'
java_import 'javax.swing.JMenuBar'
java_import 'javax.swing.JMenuItem'
java_import 'javax.swing.JOptionPane'
java_import 'javax.swing.JFileChooser'
java_import 'javax.swing.JButton'
java_import 'javax.swing.JCheckBox'
java_import 'javax.swing.JLabel'
java_import 'javax.swing.SwingConstants'
java_import 'javax.sound.midi.MidiSystem'
java_import 'javax.sound.midi.Sequencer'
java_import 'java.awt.event.ActionListener'
java_import 'java.awt.BorderLayout'
java_import 'java.awt.Font'
java_import 'java.awt.Rectangle'
java_import 'java.awt.Toolkit'
java_import 'java.net.URL'


class JMidiPlayer
  def initialize
    init
  end

  private
  def init
    midiframe = JFrame.new 'JRuby MIDI Player'
    mb = JMenuBar.new
    midiframe.set_jmenu_bar(mb)
    file = JMenu.new 'File'
    open = JMenuItem.new 'Open'
    exit = JMenuItem.new 'Exit'
    file.add(open)
    file.add(exit)
    mb.add(file)
    help = JMenu.new 'Help'
    about = JMenuItem.new 'About'
    help.add(about)
    mb.add(help)

    play = JButton.new
    status = JLabel.new
    stop = JButton.new
    loop = JCheckBox.new
    loop.setEnabled(false)

    open.add_action_listener do |e|
      @midi_file = get_file
      @midi = MidiSystem.getSequence(@midi_file)
      file_name = @midi_file.to_s.split('/')
#      puts file_name
      name = file_name[file_name.length - 1]
#      puts name
      status.setText("Status: " + name + " Loaded")
      play.setEnabled(true)
      loop.setEnabled(true)

      @sequencer = MidiSystem.getSequencer
      @sequencer.open
      @sequencer.setSequence(@midi)
    end

    def get_file
      chooser = JFileChooser.new
      @file = URL.new("file", "",
        chooser.selectedFile.getAbsolutePath)\
        if (chooser.showOpenDialog(nil)\
          == JFileChooser::APPROVE_OPTION)
    end

    exit.add_action_listener do |e|
      @sequencer.stop
      @sequencer.close
      midiframe.set_default_close_operation JFrame::DISPOSE_ON_CLOSE
      midiframe.dispose
    end

    about.add_action_listener do |e|
      JOptionPane.showMessageDialog(nil,
        "This was inspired by Rich Perilli and his JAVA MIDI player v1.0",
        "JRuby Java Swing JFrame Application.",
        JOptionPane::INFORMATION_MESSAGE)
    end

    stop.setBounds(Rectangle.new(44, 85, 75, 29))
    stop.setEnabled(false)
    stop.setFont(Font.new("Dialog", Font::BOLD, 11))
    stop.setText("Stop");
    stop.addActionListener do |e|
      status.setText("Status: Stopped")
      @sequencer.stop
      @sequencer.close
      stop.setEnabled(false)
      loop.setEnabled(true)
      play.setEnabled(true)
      open.setEnabled(true)
      about.setEnabled(true)
    end

    play.setBounds(Rectangle.new(133, 85, 75, 29))
    play.setEnabled(false)
    play.setFont(Font.new("Dialog", Font::BOLD, 11))
    play.setText("Play")
    play.addActionListener do |e|
      stop.setEnabled(true)
      play.setEnabled(false)
      loop.setEnabled(false)
      open.setEnabled(false)
      about.setEnabled(false)

      if loop.isSelected == false
        status.setText("Status: Playing (Repeat Off)")
        @sequencer.open
        @sequencer.setSequence(@midi)
      end
      if loop.isSelected == true
        @sequencer.open
        @sequencer.setSequence(@midi)
        @sequencer.setLoopCount(Sequencer::LOOP_CONTINUOUSLY)
        status.setText("Status: Playing (Repeat On)")
      end
#       status.setText("Status: Playing")

      @sequencer.start

    end

    loop.setFont(Font.new("Dialog", Font::BOLD, 11))
    loop.setText("Repeat")
    loop.setBounds(Rectangle.new(213, 90, 109, 20))
    status.setFont(Font.new("Dialog", Font::BOLD, 15))
    status.setHorizontalAlignment(SwingConstants::TRAILING)
    status.setText("Status: File Is Not Loaded Yet")
    status.setBounds(Rectangle.new(96, 41, 244, 35));

    midiframe.getContentPane.add(stop)
    midiframe.getContentPane.add(play)
    midiframe.getContentPane.add(loop)
    midiframe.getContentPane.add(status)

    # image files
    #    image1 =
Toolkit.getDefaultToolkit.getImage(URLClassLoader.getSystemResource(\
    #        "images/x.jpg")) #This will load the image file
    #    midiframe.setIconImage(image1) #this sets the icon

    midiframe.set_size(400, 200)
    midiframe.set_resizable false
    midiframe.set_default_close_operation JFrame::EXIT_ON_CLOSE
    midiframe.setVisible true

  end

end

run = JMidiPlayer.new

class AboutAction
  # Implementing a Java interface
  include ActionListener
  def action_performed(e)
    JOptionPane.showMessageDialog(nil,
      "This was inspired by Rich Perilli and his JAVA MIDI player v1.0 on Plane
Source Code",
      "JRuby Java Swing JFrame Application.",
      JOptionPane::INFORMATION_MESSAGE)
  end
end
```