```java
class MainClass
{
  public static void main(String [] args)
  {
    final int BASSDRUM = 35;
    final int SNARE = 38;
    final int CLOSEDHIHAT = 42;
    final int OPENHIHAT = 46;
    final int ENDOFSEQUENCE = 0;

    long timeZero;
    long ts=0;

    MidiWrapper midi = new MidiWrapper();

    double bpm=120; // quarter notes per minute
    int quarterNoteInterval = (int)(60000/bpm); // milliseconds

    ArrayList<RhythmEvent> sequence = new ArrayList<RhythmEvent>();

    sequence.add(new RhythmEvent(CLOSEDHIHAT,0) );
    sequence.add(new RhythmEvent(BASSDRUM,0) );

    sequence.add(new RhythmEvent(CLOSEDHIHAT,quarterNoteInterval) );
    sequence.add(new RhythmEvent(SNARE,quarterNoteInterval) );

    sequence.add(new RhythmEvent(SNARE,7*quarterNoteInterval/4) );

    sequence.add(new RhythmEvent(CLOSEDHIHAT,2*quarterNoteInterval) );
    sequence.add(new RhythmEvent(BASSDRUM,2*quarterNoteInterval) );

    sequence.add(new RhythmEvent(SNARE,9*quarterNoteInterval/4) );
    sequence.add(new RhythmEvent(BASSDRUM,10*quarterNoteInterval/4) );

    sequence.add(new RhythmEvent(CLOSEDHIHAT,3*quarterNoteInterval) );
    sequence.add(new RhythmEvent(SNARE,3*quarterNoteInterval) );

    sequence.add(new RhythmEvent(ENDOFSEQUENCE,4*quarterNoteInterval) );

    while(true)
    {
      timeZero = System.currentTimeMillis();
      ts=0;

      for(int i=0; i<sequence.size(); i++)
      {
        ts = sequence.get(i).getTimestamp();

        /*
         * While it's not time to play the next event, take a short nap
         *  to relieve the processor somewhat
         */
        while(System.currentTimeMillis() - timeZero < ts)
        {
          midi.pause(5);
        }

        if(sequence.get(i).getNote() > 0)
        {
          midi.playNote(sequence.get(i).getNote(),100);
        }
        else System.out.println("LOOP END");
      } // for
    } // while

  } // main()
} // MainClass{}
```