```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package main;

import jass.engine.SinkIsFullException;

import java.lang.reflect.Constructor;
import java.lang.reflect.InvocationTargetException;
import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.TimeUnit;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Reznov
 * @param <Voice>
 * @param <Effect>
 */
public abstract class Template<Voice, Equalizer, Compressor, Effect> {

    private String name;

    protected List<Voice> voices = new ArrayList<>();
    protected List<Equalizer> equalizers = new ArrayList<>();
    protected List<Compressor> compressors = new ArrayList<>();
    protected List<Effect> effects = new ArrayList<>();

    public Template(String name) {
        this.name = name;
    }

    public void reset() {
        voices.clear();
        equalizers.clear();
        compressors.clear();
        effects.clear();
    }

    public boolean usesCompressors() {
        return !this.compressors.isEmpty();
    }

    public String getName() {
        return this.name;
    }

    public abstract void setup(int voices, int voicesToEQAndComp, int effects, int voicesToEffects);

    public abstract void run();

    public abstract void stop();

    public abstract void tearDown();

    protected abstract void initLibrary();

}
```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package main;

import java.io.IOException;
import java.lang.management.ManagementFactory;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.Set;
import java.util.concurrent.TimeUnit;

import beads.BeadsTemplate;
import io.MeasurementWriter;
import jass.JassTemplate;
import jsyn.JSynTemplate;
import measure.Measurement;
import measure.Measurer;
import measure.Parameter;
import net.beadsproject.beads.core.io.JavaSoundAudioIO;

/**
 * @author Reznov
 */
public class StartUp {

    private final static List<Parameter> parameters = new ArrayList<>();
    private final Scanner input = new Scanner(System.in);

    static {
        /*int voices, voicesToEQandComp, effects, voicesToEffects;
        for (voices = 4; voices <= 150; voices += 2) {
            for (voicesToEQandComp = voices / 2; voicesToEQandComp <= voices; voicesToEQandComp++) {
```

```
            for (effects = 0; effects <= 5; effects++) {
                if (effects > 0) {
                    for (voicesToEffects = 1; voicesToEffects <= voices / 2; voicesToEffects++) {
                        parameters.add(new Parameter(voices, voicesToEQandComp, effects, voicesToEffects));
                    }
                } else {
                    parameters.add(new Parameter(voices, voicesToEQandComp, effects, 0));
                }
            }
        }
    }*/
    parameters.add(new Parameter(4, 4, 0, 0));
    parameters.add(new Parameter(10, 10, 1, 5));
    parameters.add(new Parameter(25, 25, 1, 10));
    parameters.add(new Parameter(40, 40, 3, 15));
    parameters.add(new Parameter(50, 50, 5, 20));
    parameters.add(new Parameter(150, 150, 0, 0));
}

private static Template[] templates = new Template[]{
        new BeadsTemplate(),
        new JassTemplate(),
        new JSynTemplate(),
};

public static void main(String[] args) throws InterruptedException, IOException {
    // new StartUp();
    run(templates[2], parameters.get(5));
    // System.out.println(System.getProperty("java.library.path"));
    // parameters.stream().filter(parameter -> parameter.getVoices() == 4).forEach(System.out::println);
    // JavaSoundAudioIO.printMixerInfo();
    // System.out.println(parameters.size());
}

private static void run(Template template, Parameter parameter) {
    template.reset();
    template.setup(parameter.getVoices(), parameter.getVoicesToEQandComp(), parameter.getEffects(), parameter.getVoicesToEffects());
    template.run();
    System.out.println(getPID());
}

public StartUp() throws InterruptedException, IOException {
    for (Template template : templates) {
        int test = 0;
        MeasurementWriter.open(template.getName());
        for (Parameter parameter : parameters) {
            template.reset();
            template.setup(parameter.getVoices(), parameter.getVoicesToEQandComp(), parameter.getEffects(), parameter.getVoicesToEffects());
            System.out.println(String.format("(%d) %s: Running parameters %s...", getPID(), template.getName(), parameter));
            TimeUnit.MILLISECONDS.sleep(1000);
            template.run();
            Measurer.measureTop(parameters.indexOf(parameter), parameter);
            // Measurer.measureSigar(parameters.indexOf(parameter), parameter);
            template.stop();
            template.tearDown();
        }
        MeasurementWriter.flush();
    }
}

public static final int getPID() {
    return Integer.valueOf(ManagementFactory.getRuntimeMXBean().getName().split("@")[0]);
}
}
```