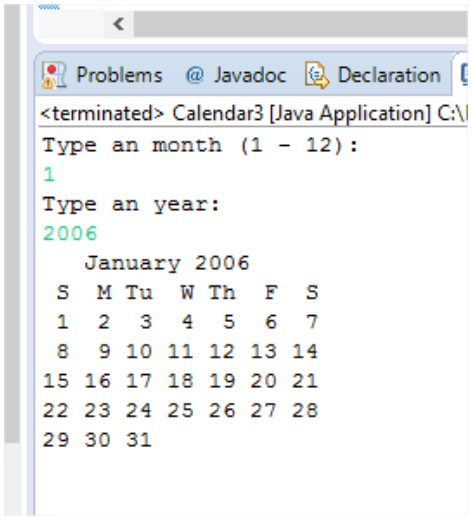
IVAN LIZCANO

1. In the site notes, you will find the code for PlayThatTuneDeluxe.java. It reads a text file sample given there to play a tune (For example, this is the file for StairwayToHeaven.txt). You are required to compile it and play that song. Once done that, modify the code to widen the spectrum to include new tones 1/8, 1/4 frequency harmonics. Now it only has 1/2 , 1 and 2 tones. Assume equal weights of 0.5.

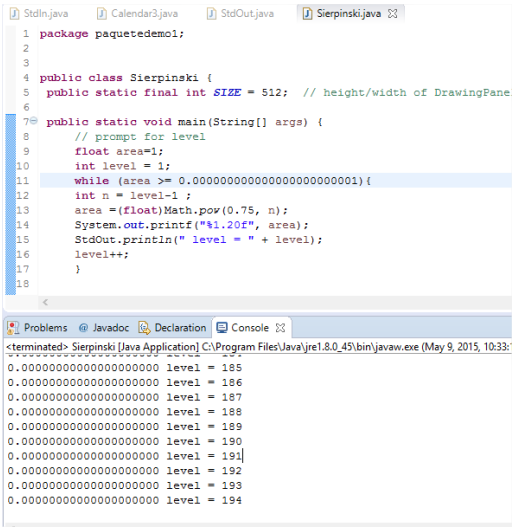
```
public static double[] note(int pitch, double t) {
    double hz = 440.0 * Math.pow(2, pitch / 12.0);
    double[] a  = tone(hz, t);
    double[] hi = tone(2*hz, t);
    double[] lo = tone(hz/2, t);
    double[] n14 = tone(hz/4, t);
    double[] n18 = tone(hz/8, t);
    double[] h  = sum(hi, lo, .5, .5);
    double[] n  = sum(n14, n18, .5, .5);
    double[] nt  = sum(h, n, .5, .5);
    return sum(a, nt, .5, .5);
   }
```

And was added 2 more clases: StdAudio and StdIn

2. Modify the Calendar.java program that takes two command-line arguments *m* and *y* and prints out the monthly calendar for the m*th* month of year *y*.

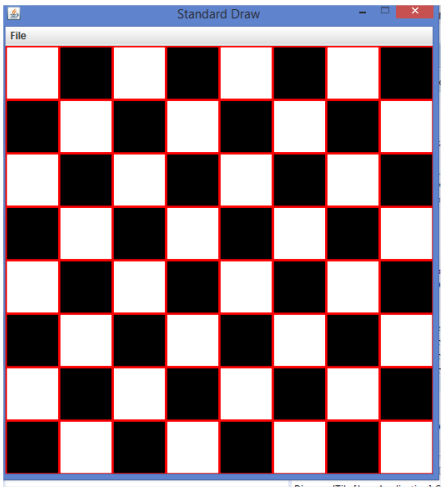


3. In the previous activity we have drawn the Sierpinski triangle using Sierpinski.java. We know from elementary school how to compute the area of a triangle. Can you come up with a static class to compute the black area of a Sierpinski Triangle? (white is the excluded area).

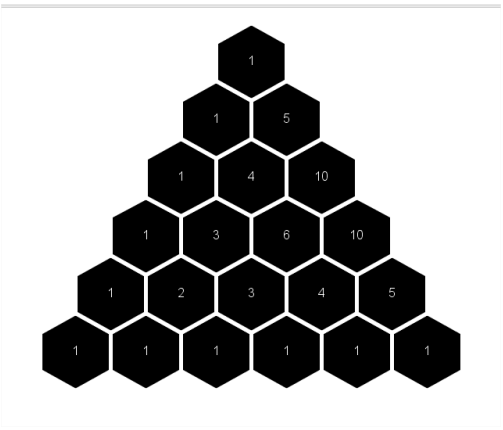


(3/4)n result the area;

4. Modify the code DiamondTile.java to produce a black and light colored chess board. You can consult the Class StdDraw to guide you thru this exercise.



5. Run the code HexTile.java to produce an hexagonal tile array. Now, modify the code to construct a Pascal triangular pyramid showing the values in the center of the hexagons. You have used Pascal.java code in Activity 2.

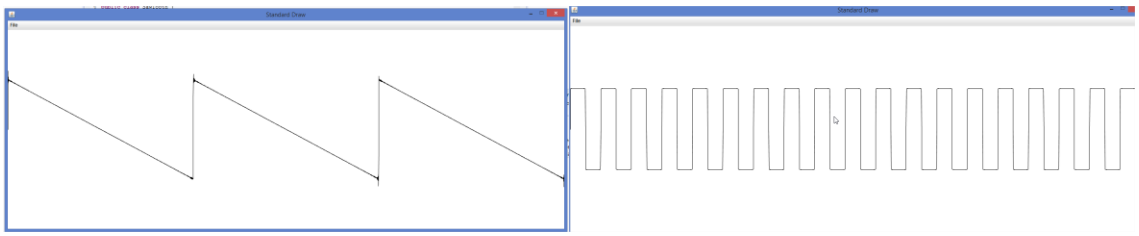


6. Write a Java static class code for checking the validity of a Visa or MasterCard credit card, using the IBM Digit Verification description.

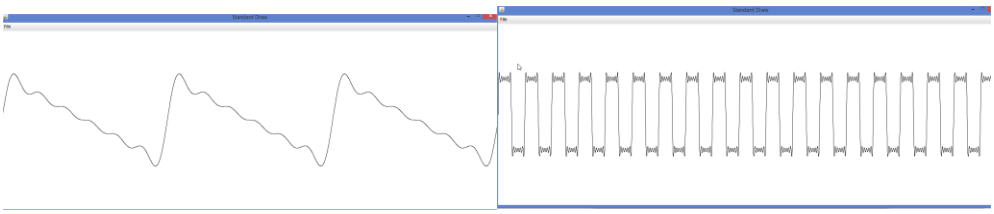
```
    public static boolean ValidCreditCard(String number){
        int s1 = 0, s2 = 0;
        String reverse = new StringBuffer(number).reverse().toString();
        for(int i = 0 ;i < reverse.length();i++){
            int digit = Character.digit(reverse.charAt(i), 10);
            if(i % 2 == 0){//this is for odd digits, they are 1-indexed in the algorith
                s1 += digit;
            }else{//add 2 * digit for 0-4, add 2 * digit - 9 for 5-9
                s2 += 2 * digit;
                if(digit >= 5){
                    s2 -= 9;
                }
            }
        }
        return (s1 + s2) % 10 == 0;
    }
}
```

7. Write a java class for approximating a square wave by a sum of N sinusoidal functions: 4/pi [sin(1*2*pi*t)/1 + sin(3*2*pi*t)/3 + sin(5*2*pi*t)/5 + ...], where pi=3.1415.... Plot it as a function of time t, and see how this sum approximates the square wave when adding more terms. This is known as Fourier Sine Series transformation. Then play the approximation using the StdAudio java class. You can follow the SawTooth.java example to approximate a sawtooth wave signal.

Sum =600



Sum = 6



8. You are invited to use the iterated function system IFS.java. This code plots an iteration algorithm for the different data. Please reproduce the following figures starting from input files: barnsley.txt, sierpinski.txt, tree.txt. Play with the files in order to reproduce different figures, different colors, etc. Produce 3 figures you like. Remember that you need a large number of points to visualize a figure, like N=10000 at least. You can read more about these figures: Barnsley fern, Sierpinski triangle, Fractal trees.