

The beast can be generated with the following code:

```
import java.util.Scanner;

public class pointCreator {

    /**
     * I have an idea, what about taking the derivative of the entire
     Bifurcation Diagram? Never mind.
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner scan = new Scanner(System.in);
        StringBuilder result = new StringBuilder();
        System.out.println("Enter a starting x value");
        double xStart = scan.nextDouble();
        int pointCount = 0;
        double y = 0;
        double y2 = 0;
        double y4 = 0;
        double y6 = 0;
        int r = 0;
        double y1 = 0;
        double[] xCoords = new double[100001];
        double[] yCoords = new double[100001];
        boolean observeable = false;
        double x;
        Plotter plot = new Plotter();
        plot.setVisible(true);
        plot.setCoordinateBounds(1, 4, 0, 1);
        for (double k = 1.01; k < 5; k += .006) {
            if (k < 3.5) {
                observeable = false;
            }
            System.out.println("K=" + k);
            y1 = 567444;
            y2 = 456;
            y4 = 4235;
            x = xStart;
            for (int a = 0; a < 100; a++) {
                y = k * (x - (x * x));
                System.out.println("Y=" + y);
            }
        }
    }
}
```

```

        if (Math.abs(y - y1) < .0001 || Math.abs(y - y2)
< .0001
        || Math.abs(y - y4) < .0001 ||
Math.abs(y - y6) < .0001) {
            observeable = true;
            a = 300;
        }
        y1 = y;
        if (a % 2 == 0) {
            y2 = y;
        }
        if (a % 4 == 0) {
            y4 = y;
        }
        if (a % 6 == 0) {
            y6 = y;
        }

        x = y;
    }
    r = 0;
    if (observeable) {

        while (r < 100) {

            y = k * (x - (x * x));
            xCoords[pointCount] = k;
            yCoords[pointCount] = y;
            System.out.println("X coord:" + k + "
Y coord:" + y);

            pointCount++;
            x = y;
            r++;
        }

    }

    plot.setPointSet(xCoords, yCoords);
}
}

```