```java
/**
 *
 * @author carlos
 */

public class Analisis_vm {

    public static void main(String[] args) {
        double[] R = new double[11];
        double[] Xo = new double[11];
        double[] R1 = new double[11];
        double[] N1 = new double[11];
        double[] X1 = new double[11];
        double[] R2 = new double[11];
        double[] N2 = new double[11];
        double[] X2 = new double[11];

        R[0] = 0.0;
        Xo[0] = 0.0;
        R1[0] = 0.0;
        N1[0] = 0.0;
        X1[0] = 0.0;
        R2[0] = 0.0;
        N2[0] = 0.0;
        X2[0] = 0.0;

        // Constantes
        final double S_1 = 0.03;
        final double S_2 = 0.1;
        final double V_1 = 7;
        final double V_2 = V_1 + 1;
        final int z = 8;
```

```java
        System.out.println("N ,   R1  ,    R2    ,       R    ,    Xo   ,   N1   ,     N2 \n");
        for (int n = 1; n <= 10; n++) {
            R1[n] = (N1[n - 1] + 1) * S_1;
            R2[n] = (N2[n - 1] + 1) * S_2;
            R[n] = (V_2 * R1[n]) + (V_1 * R2[n]);
            Xo[n] = n / (double) (z + R[n]);
            N1[n] = Xo[n] * V_2 * R1[n];
            N2[n] = Xo[n] * V_1 * R2[n];
            X1[n] = Xo[n] * V_2;
            X2[n] = Xo[n] * V_1;
            System.out.println(n + "      -   " + String.format("%.4f", R1[n]) + "  " + String.format("%.4f", R2[n])
                    + "  -   " + String.format("%.4f", R[n]) + "   " + String.format("%.4f", Xo[n])
                    + "   -   " + String.format("%.4f", N1[n]) + "  " + String.format("%.4f", N2[n]) + "\n");
        }
    }
}
```

Output - analisis_vm (run)

```
run:
N ,   R1  ,    R2    ,       R    ,    Xo   ,   N1   ,     N2

1      -   0,0300  0,1000  -   0,9400   0,1119   -   0,0268  0,0783

2      -   0,0308  0,1078  -   1,0013   0,2222   -   0,0548  0,1677

3      -   0,0316  0,1168  -   1,0705   0,3307   -   0,0837  0,2703

4      -   0,0325  0,1270  -   1,1493   0,4372   -   0,1137  0,3888

5      -   0,0334  0,1389  -   1,2394   0,5412   -   0,1446  0,5261

6      -   0,0343  0,1526  -   1,3430   0,6422   -   0,1764  0,6860

7      -   0,0353  0,1686  -   1,4626   0,7398   -   0,2089  0,8731

8      -   0,0363  0,1873  -   1,6013   0,8332   -   0,2417  1,0925

9      -   0,0373  0,2092  -   1,7628   0,9219   -   0,2747  1,3503

10      -   0,0382  0,2350  -   1,9511   1,0049   -   0,3074  1,6533

BUILD SUCCESSFUL (total time: 0 seconds)
```