```java
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.io.UnsupportedEncodingException;
import java.util.Random;

public class spatial
{
   static final double  S1       = .4;          // fox log
   static final double  S2       = 0.7;         // fox feedmax
   static final double  S3       = 0.04;        // rabbit log
   static final double  S4       = 4;           // rabbit feedmax
   static final double  S5       = 0.3;         // grass log

   static final double  a1       = 2;           // fox grow
   static final double  a2       = .8;          // fox death
   static final double  a3       = .2;          // foxes eat rabbits
   static final double  a4       = .25;         // rabbit growth
   static final double  a5       = .4;          // rabbit decay
   static final double  a6       = .39;         // grass eaten by rabbits
   static final double  a7       = .2;          // grass growth

   static final double  D_F      = 0.1;
   static final double  D_R      = .001;
   static final double  DELTA_T  = 1;
   static final double  DELTA_X  = 1;
   static final double  TTOT     = 10000;

   static double        t;
   static double[][]    fvals;
   static double[][]    rvals;
   static double[][]    gvals;
   static double[]      totalInd;
   static int           HEIGHT   = 30;
   static int           WIDTH    = 30;
   static Random        RAND     = new Random();

   public static void main(String[] args)
         throws FileNotFoundException, UnsupportedEncodingException
   {
      // Initialization
      t = 0;
      fvals = new double[HEIGHT][WIDTH];
      rvals = new double[HEIGHT][WIDTH];
      gvals = new double[HEIGHT][WIDTH];

      for (int i = 0; i < HEIGHT; i++)
         for (int j = 0; j < WIDTH; j++)
         {
            fvals[i][j] = RAND.nextDouble();
            rvals[i][j] = RAND.nextDouble() * 2;
            gvals[i][j] = RAND.nextDouble() * 10;
         }

      while (t < TTOT)
      {
         t += DELTA_T;
         double[][] tempf = new double[HEIGHT][WIDTH];
         double[][] tempr = new double[HEIGHT][WIDTH];
         double[][] tempg = new double[HEIGHT][WIDTH];

         totalInd = new double[3];

         for (int i = 0; i < HEIGHT; i++)
            for (int j = 0; j < WIDTH; j++)
            {
               // Make grid periodic/toroidal
               int nexti = 0, previ = 0, nextj = 0, prevj = 0;
               if (i == 0)
                  previ = HEIGHT - 1;
               else if (i == HEIGHT - 1)
                  nexti = 0;
               else
               {
```

```
               nexti = i + 1;
               previ = i - 1;
            }
            if (j == 0)
               prevj = WIDTH - 1;
            else if (j == WIDTH - 1)
               nextj = 0;
            else
            {
               nextj = j + 1;
               prevj = j - 1;
            }
            // Calculate increment
            tempf[i][j] = fvals[i][j] + DELTA_T * (D_F / (DELTA_X * DELTA_X)
                  * (fvals[i][nextj] + fvals[i][prevj] + fvals[nexti][j]
                        + fvals[previ][j] - 4 * fvals[i][j])
                  + a1 * fvals[i][j] * rvals[i][j] * (1 - S1 * fvals[i][j])
                        / (1 + S2 * rvals[i][j])
                  - a2 * fvals[i][j]);
            totalInd[0] += tempf[i][j];

            tempr[i][j] = rvals[i][j] + DELTA_T * (D_R / (DELTA_X * DELTA_X)
                  * (rvals[i][nextj] + rvals[i][prevj] + rvals[nexti][j]
                        + rvals[previ][j] - 4 * rvals[i][j])
                  - a3 * fvals[i][j] * rvals[i][j] * (1 - S1 * fvals[i][j])
                        / (1 + S2 * rvals[i][j])
                  + a4 * rvals[i][j] * gvals[i][j] * (1 - S3 * rvals[i][j])
                  - a5 * rvals[i][j]);
            totalInd[1] += tempr[i][j];

            tempg[i][j] = gvals[i][j]
                  + DELTA_T * (-a6 * rvals[i][j] * gvals[i][j]
                        * (1 - S3 * rvals[i][j]) / (1 + S4 * gvals[i][j])
                        + a7 * gvals[i][j] * (1 - S5 * gvals[i][j]));
            totalInd[2] += tempg[i][j];
         }
      fvals = tempf;
      rvals = tempr;
      gvals = tempg;
      System.out.println(t + "\t" + totalInd[0] + "\t" + totalInd[1] + "\t"
            + totalInd[2]);

      // Print output.
      if ((int) t % 2000 == 0)
         print();
   }
}

// Increment size of map diagonally (= add cross in toroidal case)
public static double[][][] expandDiag(double[][] matrixOne,
      double[][] matrixTwo)
{

   double[][] newMatrixOne = new double[matrixOne.length
         + 1][matrixOne[0].length + 1];
   double[][] newMatrixTwo = new double[matrixTwo.length
         + 1][matrixTwo[0].length + 1];

   for (int i = 0; i < HEIGHT; i++)
      for (int j = 0; j < WIDTH; j++)
      {
         newMatrixOne[i][j] = matrixOne[i][j];
         newMatrixTwo[i][j] = matrixTwo[i][j];
      }

   for (int i = 0; i < newMatrixTwo.length; i++)
   {
      newMatrixOne[newMatrixOne.length - 1][i] = Double.MIN_VALUE;
      newMatrixTwo[newMatrixTwo.length - 1][i] = Double.MIN_VALUE;
      newMatrixOne[i][newMatrixOne.length - 1] = Double.MIN_VALUE;
      newMatrixTwo[i][newMatrixTwo.length - 1] = Double.MIN_VALUE;
   }
   HEIGHT += 1;
```

```java
        WIDTH += 1;
        double[][][] newMatrices = new double[2][HEIGHT][WIDTH];
        newMatrices[0] = newMatrixOne;
        newMatrices[1] = newMatrixTwo;

        return newMatrices;
    }

    public static void print()
            throws FileNotFoundException, UnsupportedEncodingException
    {
        PrintWriter writer = new PrintWriter(
                "/home/william/b16_henrikahl/popdyn/t" + (int) t + ".dat", "UTF-8");
        for (int i = 0; i < HEIGHT; i++)
        {
            for (int j = 0; j < WIDTH; j++)
            {
                writer.println(t + "\t" + i + "\t" + j + "\t" + fvals[i][j] + "\t"
                        + rvals[i][j] + "\t" + gvals[i][j]);
            }
            writer.println();
        }
        writer.close();
    }

    public static void printEvery()
            throws FileNotFoundException, UnsupportedEncodingException
    {
        PrintWriter totWriter = new PrintWriter(
                "/home/william/b16_henrikahl/popdyn/tot_t" + (int) t + ".dat",
                "UTF-8");
        totWriter.println(
                t + "\t" + totalInd[0] + "\t" + totalInd[1] + "\t" + totalInd[2]);
        totWriter.close();
    }
}
```