```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Weather
{

    class Program
    {
        static double[][] defaultProbabilities = new double[][]
        {                                       //                              tomorrow:
                                                //                      Sunny   Cloudy   Raining
            new double[]{ 0.8, 0.2, 0.0 },      //          Sunny       0.8     0.2      0.0
            new double[]{ 0.4, 0.4, 0.2 },      // today:   Cloudy      0.4     0.4      0.2
            new double[]{ 0.2, 0.6, 0.2 }       //          Raining     0.2     0.6      0.2
        };

        static void Main(string[] args)
        {
            WeatherGenerator.generateAndPrintRandomSequence(
                    defaultProbabilities, WeatherGenerator.WeatherType.Sunny, 100);
            Console.ReadKey();
        }
    }


    class WeatherGenerator
    {
        public enum WeatherType { Sunny, Cloudy, Raining };
        static int typesCount = Enum.GetNames(typeof(WeatherType)).Length;


        public static void generateAndPrintRandomSequence(
                    double[][] probabilities, WeatherType todayWeather, int countOfDays)

        {
            Console.Out.WriteLine("Today weather:\n    {0}", todayWeather);
            Console.Out.WriteLine("Next {0} days:", countOfDays);
            double[][] cummulativeProbabilities=initCummulativeProbabilities(probabilities);
            generateRandomSequence(cummulativeProbabilities, todayWeather, countOfDays);
        }


        private static double[][] initCummulativeProbabilities(
                    double[][] probabilities)
        {
            double[][] cummulativeProbabilities = new double[typesCount][];
            for (int i = 0; i < typesCount; i++)
            {
                double sum = 0;
                double[] cummProb = new double[typesCount];
                for (int j = 0; j < typesCount; j++)
                {
                    sum = sum + probabilities[i][j];
                    cummProb[j] = sum;
                }
                cummulativeProbabilities[i] = cummProb;
            }
            return cummulativeProbabilities;
        }
```

```
        private static void generateRandomSequence(
              double[][] cummulativeProbabilities,
              WeatherType todayWeather,
              int countOfDays)
        {
            Random rand = new Random();
            WeatherType currentWeather = todayWeather;
            for (int k = 1; k <= countOfDays; k++)
            {
                double r = rand.NextDouble();                // 0 < r < 1
                for (int j = 0; j < typesCount; j++)
                {
                    if (r < cummulativeProbabilities[(int)currentWeather][j])
                    {
                        currentWeather = (WeatherType)Enum.GetValues(typeof(WeatherType))
                                                .GetValue(j);
                        Console.Out.WriteLine
                            ("{0,-3}:{1}", k, currentWeather);
                        break;
                    }
                }
            }
        }
    }
}
```