

Triângulo de Pascal

Mário Leite

...

Na Matemática existem assuntos muito interessantes agrupados e que, às vezes, nos passam despercebidos, mas que estão entrelaçados. Um desses assuntos é o famoso “Triângulo de Pascal”; na verdade uma sequência de números inteiros que podem ser mostrados na forma de um triângulo estilizado, formando uma estrutura matemática bem definida e organizada, onde cada número é a soma dos dois números diretamente acima dele na linha anterior. E embora possa ser gerado independentemente de outras estruturas matemáticas, ele tem ligação com outros ramos da Matemática e pode ser utilizado em situações práticas que requerem estruturas como esta.

Apenas como exemplo, vale citar algumas situações que essa estrutura pode ser usada: “Sequência de Fibonacci” em que a soma dos números de sua diagonal gera essa sequência. Na expansão de binômios onde os coeficientes dos termos na expansão de $(a+b)^n$ pode ser usada para calcular combinações, que são fundamentais na teoria das probabilidades. Para calcular a probabilidade de obter um certo número de sucessos em uma série de experimentos de Bernoulli (Dinâmica dos Fluidos). No cálculo de combinações na “Teoria dos Jogos” para encontrar padrões e estratégias vencedoras. Na “Engenharia” e “Ciência da Computação” com algoritmos de geração de subconjuntos. Na “Análise Combinatória” para ajudar a resolver problemas relacionados a arranjos e permutações. Na “Modelagem de Crescimento de Populações” e “Processos Estocásticos” para descrever probabilidades de diferentes eventos. Na “Economia” em modelos de decisão econômica e análise de risco onde coeficientes binomiais podem ser usados para calcular probabilidades e expectativas, etc.

Portanto, muito além de ser um simples exemplo de exercício para programadores iniciantes, o “Triângulo de Pascal” é uma ferramenta que deve ser levada em consideração na hora de resolver problemas práticos que envolvem estruturas numéricas. O programa “**TrianguloPascal**”, codificado em C#, é uma solução simples que mostra essa estrutura estilizada num triângulo retângulo formado pelos elementos abaixo da diagonal principal, seguindo a ordem de apresentação de linhas em cores: com as linhas ímpares na cor amarela e as linhas pares na cor verde.

```
D:\Cantinho da Programação\Códigos\CSharp\TrianguloPascal\TrianguloPascal\bin\Debug\TrianguloPascal.exe
Digite o número de linhas do triângulo [de 2 a 20]: 20

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1
1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1
1 15 105 455 1365 3003 5005 6435 6435 5005 3003 1365 455 105 15 1
1 16 120 560 1820 4368 8008 11440 12870 11440 8008 4368 1820 560 120 16 1
1 17 136 680 2380 6188 12376 19448 24310 24310 19448 12376 6188 2380 680 136 17 1
1 18 153 816 3060 8568 18564 31824 43758 48620 43758 31824 18564 8568 3060 816 153 18 1
1 19 171 969 3876 11628 27132 50388 75582 92378 92378 75582 50388 27132 11628 3876 969 171 19 1
```

```

using System;

namespace TrianguloPascal
{
    class TrianguloPascal
    {
        static void Main()
        {
            int n = 0;
            while (n < 2 || n > 20)
            {
                Console.WriteLine("Digite o número de linhas do triângulo [de 2 a 20]: ");
                n = Math.Abs(int.Parse(Console.ReadLine()));
                Console.WriteLine("");
            }
            // Inicia a matriz para armazenar os elementos
            int[,] LstPascal = new int[n, n];
            // Preenche a matriz com os valores do triângulo de Pascal
            for (int lin = 0; lin < n; lin++)
            {
                for (int col = 0; col <= lin; col++)
                {
                    if (col == 0 || lin == col)
                    {
                        LstPascal[lin, col] = 1;
                    }
                    else
                    {
                        LstPascal[lin,col]=LstPascal[lin-1, col] + LstPascal[lin-1, col-1];
                    }
                }
            }
            // Imprime o triângulo de Pascal com cores
            for (int lin = 0; lin < n; lin++)
            {
                // Define a cor da linha
                if (lin % 2 == 0)
                {
                    Console.ForegroundColor = ConsoleColor.Yellow;
                }
                else
                {
                    Console.ForegroundColor = ConsoleColor.Green;
                }

                for (int col = 0; col <= lin; col++)
                {
                    // Imprime elemento de acordo com o seu tamanho
                    if (LstPascal[lin, col] < 10)
                    {
                        Console.WriteLine(LstPascal[lin, col] + "  ");
                    }
                    else if (10 <= LstPascal[lin, col] && LstPascal[lin, col] < 100)
                    {
                        Console.WriteLine(LstPascal[lin, col] + " ");
                    }
                    else
                    {
                        Console.WriteLine(LstPascal[lin, col] + " ");
                    }
                }
                Console.WriteLine(); // Salta para uma nova linha do triângulo
            }
            // Reseta a cor para a cor padrão
            Console.ReadKey();
            Console.ResetColor();
        }
    }
}

```