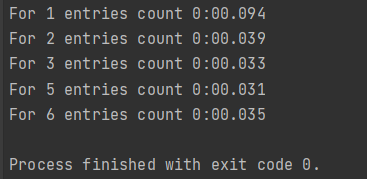
# Лабораторная работа 2

Программа заполняет стадион, состоящий из нескольких секций с определенным количеством мест в каждой, людьми. Программа выводит время, необходимое для заполнения стадиона, с разным количеством порталов для входа в него.

## Вывод программы:



## Код программы:

**Program.cs**

internal static class Program

{

private const int STADIUM\_SECTIONS\_COUNT = 30;

private const int STADIUM\_SITTINGS\_COUNT = 100000;

private const int MAX\_STADIUM\_ENTRIES\_COUNT = 6;

private static void Main()

{

for (var i = 1; i <= MAX\_STADIUM\_ENTRIES\_COUNT; i++)

{

if (STADIUM\_SECTIONS\_COUNT % i != 0) continue;

var stopwatch = new Stopwatch();

stopwatch.Start();

FillStadium(InitStadium(STADIUM\_SECTIONS\_COUNT, STADIUM\_SITTINGS\_COUNT), STADIUM\_SECTIONS\_COUNT,

STADIUM\_SITTINGS\_COUNT, i);

Console.Write($"For {i} entries count ");

stopwatch.Stop();

var timeTaken = stopwatch.Elapsed;

Console.WriteLine(timeTaken.ToString(@"m\:ss\.fff"));

}

}

private static int[][] InitStadium(int sectionsCount, int sittingsCount)

{

var stadium = new int[sectionsCount][];

for (var i = 0; i < sectionsCount; i ++)

{

stadium[i] = new int[sittingsCount];

for (var j = 0; j < sittingsCount; j ++)

{

stadium[i][j] = 0;

}

}

return stadium;

}

private static void FillSections(int[][] stadium, int sectionsCount, int sittingsCount, int stadiumEntriesCount,

int entryIndex)

{

for (var i = entryIndex \* (sectionsCount / stadiumEntriesCount);

i < entryIndex \* (sectionsCount / stadiumEntriesCount) + sectionsCount / stadiumEntriesCount;

i++)

{

for (var j = 0; j < sittingsCount; j++)

{

stadium[i][j] = 1;

}

}

}

private static void FillStadium(int[][] stadium, int sectionsCount, int sittingsCount, int stadiumEntriesCount)

{

Parallel.For(

0,

stadiumEntriesCount,

i =>

{

FillSections(stadium, sectionsCount, sittingsCount, stadiumEntriesCount, i);

});

}

}