**14. Многопоточные приложения**

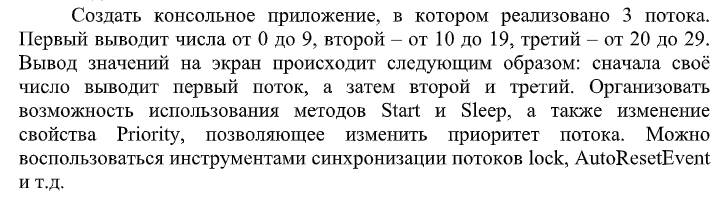
Задание 1.

Рисунок 14.1 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.Threading;

using System.Diagnostics;

namespace ThreadingClass

{

class Program

{

public static void Main(string[] args)

{

Thread thread1 = new Thread(new ThreadStart(FirstThread));

Thread thread2 = new Thread(new ThreadStart(SecondThread));

Thread thread3 = new Thread(new ThreadStart(ThirdThread));

thread1.Start();

thread1.Join();

thread2.Start();

thread2.Join();

thread3.Start();

thread3.Join();

}

static void FirstThread()

{

Console.WriteLine("Первый поток ---");

for (int i = 0; i <= 9; i++)

{

Console.WriteLine(i);

}

}

static void SecondThread()

{

Console.WriteLine("Второй поток ---");

for (int i = 10; i < 20; i++)

{

Console.WriteLine(i);

}

}

static void ThirdThread()

{

Console.WriteLine("Третий поток ---");

for (int i = 20; i <= 39; i++)

{

Console.WriteLine(i);

}

}

}

}

Таблица 14.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Числа | 9-30 |

Анализ результатов:

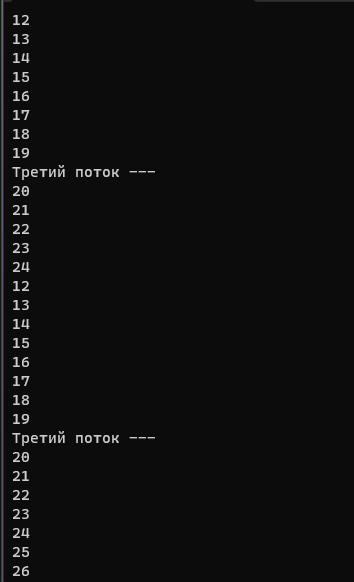


Рисунок 14.2 – Результат работы программы

Источник: собственная разработка

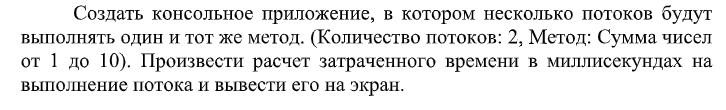
Задание 2.

Рисунок 14.3 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Diagnostics;

namespace zad2

{

internal class Program

{

public static void Main(string[] args)

{

Thread thread1 = new Thread(new ThreadStart(PotokOperciy));

thread1.Start();

thread1.Join();

}

static void PotokOperciy() {

Stopwatch sWatch = new Stopwatch();

sWatch.Start();

int step = 0;

Console.WriteLine("Potok is Run!!!!!!");

for (int i = 0; i <= 10; i++)

{

step += i;

}

Console.WriteLine(step);

sWatch.Stop();

Console.WriteLine(sWatch.ElapsedMilliseconds.ToString());

}

}

}

Таблица 14.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Числа | Сумма чисел |

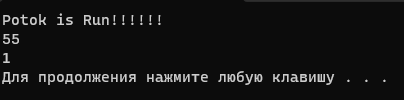
Анализ результатов:

Рисунок 14.4 – Результат работы программы

Источник: собственная разработка

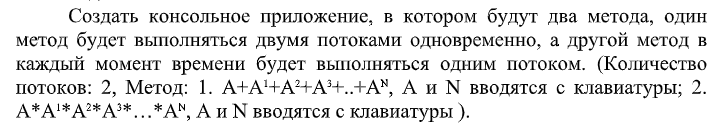
Задание 3.

Рисунок 14.5 – Условие задания

Источник: собственная разработка

Листинг программы:

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace zad3

{

internal class Program

{

public static void Main(string[] args)

{

Thread thread1 = new Thread(new ThreadStart(First));

Thread thread2 = new Thread(new ThreadStart(Second));

thread1.Start();

thread1.Join();

thread2.Start();

thread2.Join();

}

static void First()

{

Console.WriteLine("Введите А: ");

int A = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите N: ");

int N = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Первый поток");

int item = 0;

for (int i = A; i < N; i++)

{

item += Convert.ToInt32(Math.Pow(A, N));

}

Console.WriteLine(item);

}

static void Second()

{

Console.WriteLine("Введите А: ");

int A = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите N: ");

int N = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Второй поток");

int item = 1;

for (int i = A; i < N; i++)

{

item += item \* Convert.ToInt32(Math.Pow(A, N));

}

Console.WriteLine(item);

}

}

}

Таблица 14.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| А = 1, N = 5; А = 1, N = 8 | Первый = 4, Второй = 128 |

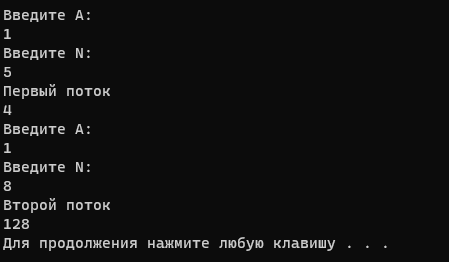
Анализ результатов:

Рисунок 14.6 – Результат работы программы

Источник: собственная разработка