**Выполнили студент 215 группы**

**Марков К.А.**

**Практическое занятие 18**

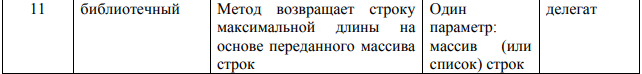
**Тема: «Применение класса THREAD»**

**Цель работы:** научиться использовать базовые возможности класса потоков Thread.

**Задание:**

1. Создайте консольное приложение C# в MS Visual Studio. 26

2. Реализуйте решение задачи в соответствии с индивидуальным вариантом с использованием асинхронного вызова метода. При реализации получения результата работы асинхронного метода используйте механизм обратного асинхронного вызова.



class Program

{

// Размерность матрицы

const int Rows = 5;

const int Cols = 5;

// Cоздается объект для блокировки доступа к консоли.

// Этот объект используется для синхронизации доступа к консоли из разных потоков.

static object consoleLock = new object();

static void Main(string[] args)

{

// Создаем массив потоков

Thread[] threads = new Thread[2];

// Первый поток генерирует и выводит исходную матрицу

threads[0] = new Thread(() =>

{

// Захватываем блокировку для вывода сообщения и матрицы

lock (consoleLock)

{

Console.WriteLine("Исходная матрица:");

int[,] matrix = GenerateRandomMatrix(); // Генерация исходной матрицы

PrintMatrix(matrix); // Вывод исходной матрицы

Console.WriteLine(); // Пустая строка для отделения матриц

}

});

// Второй поток генерирует, преобразует и выводит преобразованную матрицу

threads[1] = new Thread(() =>

{

// Захватываем блокировку для вывода сообщения и матрицы

lock (consoleLock)

{

Console.WriteLine("Преобразованная матрица:");

int[,] matrix = GenerateRandomMatrix(); // Генерация исходной матрицы

TransformMatrix(matrix); // Преобразование матрицы

PrintMatrix(matrix); // Вывод преобразованной матрицы

Console.WriteLine(); // Пустая строка для отделения матриц

}

});

// Запускаем каждый поток на выполнение и ожидаем их завершения

foreach (var thread in threads)

{

thread.Start();

thread.Join();

}

// Выводим сообщение о завершении работы главного потока

Console.WriteLine("Главный поток завершил свою работу.");

Console.ReadKey();

}

// Метод для генерации случайной матрицы

static int[,] GenerateRandomMatrix()

{

Random rand = new Random();

int[,] matrix = new int[Rows, Cols];

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

{

for (int j = 0; j < Cols; j++)

{

matrix[i,j] = rand.Next(1, 101); // Случайные числа от 1 до 100

}

}

return matrix;

}

// Метод для преобразования матрицы: нечетные элементы заменяем на -1

static void TransformMatrix(int[,] matrix)

{

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

{

for (int j = 0; j < Cols; j++)

{

if (matrix[i, j] % 2 != 0)

matrix[i, j] = -1;

}

}

}

// Метод для вывода матрицы в консоль

static void PrintMatrix(int[,] matrix)

{

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

{

for (int j = 0; j < Cols; j++)

{

Console.Write(matrix[i, j] + "\t");

}

Console.WriteLine();

}

}

}

**Вывод:** научился использовать базовые возможности класса потоков Thread.