

Лабораторная работа №22

Асинхронное программирование

1 Цель работы

- 1.1 Научиться реализовывать и запускать асинхронные операции на C#.
- 1.2 Научиться выполнять вычисления, используя асинхронные операции.
- 1.3 Научиться выполнять ввод и вывод данных, используя асинхронные операции.

2 Литература

- 2.1 <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/async/>
- 2.2 <https://metanit.com/sharp/tutorial/> - гл. 16

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание и асинхронный вызов метода, не возвращающего результат

5.1.1 Создать в консольном приложении метод для вычисления a в степени x (a – любое, x – целое). Вычисление реализовать без использования стандартной математической функции. Метод должен выводить результат вычислений на экран в следующем виде (вместо переменных должны быть значения переменных):

$a^x = \text{result}$

5.1.2 Вызвать метод для вычисления a^x , используя последовательный асинхронный вызов для трех различных наборов параметров методов.

5.1.3 Вызвать метод для вычисления a^x , используя параллельный асинхронный вызов для трех различных наборов параметров методов.

- 5.2 Создание и асинхронный вызов метода, возвращающего результат

5.2.1 Создать в консольном приложении метод для вычисления a в степени x (a – любое, x – целое). Вычисление реализовать без использования стандартной математической функции. Метод должен возвращать результат вычислений.

5.2.2 Вызвать метод для вычисления a^x , используя параллельный асинхронный вызов для трех различных наборов параметров методов. Полученные результаты вывести на экран.

- 5.2.3 Создать в консольном приложении асинхронный метод, вычисляющий:

$(a1^x1 + a2^x2) / (a3^x3 - a4^x4)$

Для вычислений a^x использовать последовательный вызов метода a^x .

- 5.3 Реализация асинхронной записи в файл

5.3.1 Создать в консольном приложении асинхронный метод, выполняющий запись N случайных целых чисел в файл (параметры метода: имя файла и количество строк). N должно быть большим (100 000 – 1 000 000).

Образец заполнения файла:

Число №1: 156

Число №2: 269

Число №3: 1278

...

Для записи можно использовать асинхронный метод `WriteLineAsync` класса `StreamWriter`:

```
using (StreamWriter writer = new StreamWriter(имя файла, false))
{
    await writer.WriteLineAsync(строка); // асинхронная запись в файл
}
```

В начале метода написать, что запись в файл `filename` начата, в конце метода – что закончена.

5.3.2 Вызвать метод записи файла в асинхронном режиме и после него вывести строку «Конец программы».

5.4 Реализация асинхронного чтения из файла

5.4.1 Создать в консольном приложении асинхронный метод, выполняющий построчное чтение данных из файла (параметры метода: имя файла).

Для построчного чтения можно использовать асинхронный метод `ReadLineAsync` класса `StreamReader`:

```
using (StreamReader reader = new StreamReader(имя файла))
{
    string line;
    while ((line = await reader.ReadLineAsync()) != null) // асинхронное чтение строки
    {
        ...
    }
}
```

В начале метода написать, что чтение из файла `filename` начато, в конце метода – что закончено.

Каждую строку из файла выводить в следующем виде:

имя файла: считанная строка

5.4.2 Вызвать метод чтения файла в асинхронном параллельном режиме для трех разных объемных файлов (можно использовать файлы, созданные в п.5.3). Полученные данные вывести на экран.

5.5 Отмена длительной операции

5.5.1 Реализовать отмену длительной операции (чтения или записи файла) по истечении определенного времени.

5.5.2 Реализовать перехват исключений при появлении исключительной ситуации.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C#. Выполнить все задания из п.5 в одном решении LabWork22.

При разработке считать, что пользователь ввел данные требуемого типа, остальные возможные ошибки обрабатывать.

При выполнении заданий использовать минимально возможное количество команд и переменных и выполнять форматирование и рефакторинг кода.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие ключевые слова используются в C# для работы с асинхронными вызовами?

8.2 Какие типы возврата могут быть у асинхронных методов и для чего предназначен каждый из типов?

8.3 Как вызвать метод в асинхронном режиме?

8.4 Как указать, что в методе могут быть асинхронные вызовы?

8.5 Как обработать исключения, возникшие в асинхронных вызовах