

# Практическая работа №20

## Реализация ввода-вывода методами асинхронного программирования

### 1 Цель работы

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

### 2 Литература

- 2.1 <https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/concepts/async/>
- 2.2 <https://metanit.com/sharp/tutorial/> - гл.15-16

### 3 Задание

#### 3.1 Реализация записи в файл

3.1.1 Создать в консольном приложении метод, выполняющий запись N случайных 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 указанный файл начата, в конце метода – что закончена.

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

#### 3.2 Реализация чтения из файла

3.2.1 Создать в консольном приложении метод, выполняющий чтение данных из файла и возвращающий результат (параметры метода: имя файла).

Для чтения можно использовать асинхронный метод `ReadToEndAsync` класса `StreamReader`:

```
using (StreamReader reader = new StreamReader(имя файла))  
{  
    string result = await reader.ReadToEndAsync(); // асинхронное чтение из файла  
}
```

В начале метода написать, что чтения из указанного файла начато, в конце метода – что закончено.

3.2.2 Вызвать метод чтения файла в асинхронном режиме для трех разных объемных файлов (можно использовать файлы, созданные в п.3.1). Полученные данные вывести на экран.

3.3 Добавить обработку исключений, возникающих при асинхронном чтении и записи файлов.

3.4 Добавить прерывание при записи файла дольше  $n$  миллисекунд, используя `CancellationTokenSource`. Добавить обработку исключения `TaskCanceledException` и вывод информации о том, что превышено время выполнения операции.

#### **4 Порядок выполнения работы**

4.1 Выполнить все задания из п.3 в одном решении `PractWork20`. Возможные ошибки требуется обрабатывать. Выполнить форматирование и рефакторинг кода.

4.2 Ответить на контрольные вопросы.

#### **5 Содержание отчета**

5.1 Титульный лист

5.2 Цель работы

5.3 Ответы на контрольные вопросы

5.4 Вывод

#### **6 Контрольные вопросы**

6.1 Каков шаблон имени асинхронного метода?

6.2 Как отметить выполнение асинхронного метода?

6.3 Как выполнить асинхронное чтение данных из файла?

6.4 Как выполнить асинхронную запись данных в файл?