**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БАШКОРТОСТАН**

Государственное бюджетное профессиональное образовательноеучреждение

«Уфимский колледж статистики, информатики и вычислительной техники»

Реферат

Тема: Библиотеки C#, пространство имен System.Threading.

Дисциплина: Прикладное программирование

Выполнил:

Студент гр. 16П-2

\_\_\_\_\_\_\_\_\_\_\_ Батурин Н. С.

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018г.

Проверил преподаватель

\_\_\_\_\_\_\_\_\_\_\_Зиганшина Р.С.

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018г.

Уфа 2018

Введение

В языке программирования C# можно подключать различные библиотеки путем добавления ссылки на них в обозревателе решений, а также добавив в проект строчку «using <название пространства имен>;». Но некоторые библиотеки включены в сборку по умолчанию, такие как System, System.Text, System.IO, System.Linq… Вот о такой библиотеке, как вышеперечисленные, и пойдет речь.

Пространство имен System.Threading

В библиотеке System.Core существует пространство имен System.Threading, которое содержит классы и интерфейсы, которые дают возможность программировать в многопоточном режиме. Помимо классов синхронизации работы потоков и доступа, это пространство имен содержит класс ThreadPool, позволяющий использовать пул предоставляемых системой потоков, и класс Timer, выполняющий методы обратного вызова в потоках пула потоков.

Классы

AbandonedMutexException – Исключение вызывается, когда некоторый поток получает объект Mutex, брошенный другим потоком путем выхода без высвобождения.

AsyncLocal<T> - Представляет внешние данные, локальные для данного асинхронного потока управления, такие как асинхронный метод.

AutoResetEvent - Уведомляет ожидающий поток о том, что произошло событие. Этот класс не наследуется.

Barrier - Позволяет нескольким задачам параллельно работать с алгоритмом, используя несколько фаз.

BarrierPostPhaseException - Исключение, которое возникает при сбое действия барьера Barrier, выполняемого в конце фазы

CancellationTokenSource - Отправляет токену CancellationToken сигнал отмены.

CompressedStack - Обеспечивает методы для установки и инициализации сжатого стека в текущем потоке. Этот класс не наследуется.

CountdownEvent - Представляет примитив синхронизации, которому отправляется сигнал при достижении счетчиком нуля.

EventWaitHandle - Представляет событие синхронизации потока.

ExecutionContext - Управляет контекстом выполнения текущего потока. Этот класс не наследуется.

HostExecutionContext - Инкапсулирует и распространяет контекст выполнения хоста по потокам.

HostExecutionContextManager - Обеспечивает функциональность, позволяющие хосту общеязыковой среды выполнения участвовать в перемещении или миграции контекста выполнения.

Interlocked - Предоставляет атомарные операции для переменных, общедоступных нескольким потокам.

LazyInitializer - Обеспечивает процедуры инициализации адаптирующегося типа.

LockRecursionException - Исключение, которое создается, когда рекурсивная запись блокировки не совместима с рекурсивной политикой блокировки.

ManualResetEvent - Уведомляет один или более ожидающих потоков о том, что произошло событие. Этот класс не наследуется.

ManualResetEventSlim - Предоставляет уменьшенную версию ManualResetEvent.

Monitor - Предоставляет механизм для синхронизации доступа к объектам.

Mutex - Примитив синхронизации, который также может использоваться в межпроцессорной синхронизации.

Overlapped - Предоставляет управляемое представление структуры OVERLAPPED Win32, включая методы для передачи данных из экземпляра Overlapped в структуру NativeOverlapped.

PreAllocatedOverlapped - Представляет предварительно выделенное состояние для собственных перекрывающихся операций ввода-вывода.

ReaderWriterLock - Определяет блокировку, которая поддерживает один пишущий поток и несколько читающих.

ReaderWriterLockSlim - Представляет блокировку, используемую для управления доступом к ресурсу, которая позволяет нескольким потокам производить считывание или получать монопольный доступ на запись.

RegisteredWaitHandle - Представляет дескриптор, который регистрируется при вызове метода RegisterWaitForSingleObject(WaitHandle, WaitOrTimerCallback, Object, UInt32, Boolean). Этот класс не наследуется.

Semaphore - Ограничивает число потоков, которые могут одновременно обращаться к ресурсу или пулу ресурсов.

SemaphoreFullException - Исключение, выдаваемое при вызове метода Release для семафора, значение счетчика которого уже равно максимальному.

SemaphoreSlim - Представляет упрощенную альтернативу семафору Semaphore, ограничивающему количество потоков, которые могут параллельно обращаться к ресурсу или пулу ресурсов.

SynchronizationContext - Обеспечивает базовую функциональность для распространения контекста синхронизации в различных моделях синхронизации.

SynchronizationLockException - Исключение, которое создается в то время, когда методу требуется вызвавший его объект для получения блокировки данного монитора, а метод вызван объектом, не являющимся владельцем блокировки.

Thread - Создает и контролирует поток, задает приоритет и возвращает статус.

ThreadAbortException - Исключение, выдаваемое при вызове метода Abort(Object). Этот класс не наследуется.

ThreadExceptionEventArgs - Предоставляет данные для события ThreadException.

ThreadInterruptedException - Исключение, которое выдается при прерывании работы потока Thread, находящегося в состоянии ожидания.

ThreadLocal<T> - Предоставляет хранилище для данных, локальных для потока.

ThreadPool - Предоставляет пул потоков, который можно использовать для выполнения задач, отправки рабочих элементов, обработки асинхронного ввода-вывода, ожидания от имени других потоков и обработки таймеров.

ThreadPoolBoundHandle - Представляет дескриптор ввода-вывода, который привязан к системному пулу потоков и позволяет низкоуровневым компонентам получать уведомления для асинхронных операций ввода-вывода.

ThreadStartException - Исключение выдается, когда происходит сбой в управляемом потоке после запуска базового потока операционной системы, но до готовности потока к выполнению кода пользователя.

ThreadStateException - Исключение, которое выдается, когда объект Thread является неправильным свойством ThreadState для вызова метода.

Timeout - Содержит константы, определяющие бесконечные интервалы времени ожидания. Этот класс не наследуется.

Timer - Предоставляет механизм для выполнения метода в потоке пула с заданными интервалами. Этот класс не наследуется.

Volatile - Содержит методы для выполнения операций энергонезависимой памяти.

WaitHandle - Инкапсулирует связанные с операционной системой объекты, ожидающие монопольного доступа к общим ресурсам.

WaitHandleCannotBeOpenedException - Исключение, которое возникает при попытке открыть несуществующий системный семафор или дескриптор ожидания события.

WaitHandleExtensions - Предоставляет удобные методы для работы с безопасным дескриптором для дескриптора ожидания.

Структуры

AsyncFlowControl - Обеспечивает функциональность для восстановления миграции или перемещения контекста выполнения между потоками.

AsyncLocalValueChangedArgs<T> - Класс, предоставляющий сведения об изменениях данных экземплярам AsyncLocal<T>, которые зарегистрированы для получения уведомлений об изменениях.

CancellationToken - Распространяет уведомление о том, что операции следует отменить.

CancellationTokenRegistration - Представляет делегат обратного вызова, зарегистрированный с объектом CancellationToken.

LockCookie - Определяет блокировку, которая реализует семантику "один записывающий / много читающих". Это тип значения.

NativeOverlapped - Содержит явно заданный макет, видимый из неуправляемого кода и имеющий тот же макет, что и структура OVERLAPPED Win32, с дополнительными зарезервированными полями в конце.

SpinLock - Предоставляет примитив взаимно исключающей блокировки, в котором поток, пытающийся получить блокировку, ожидает в состоянии цикла, проверяя доступность блокировки.

SpinWait - Предоставляет поддержку ожидания на основе прокруток.

Перечисления

ApartmentState - Задает апартаментное состояние потока Thread.

EventResetMode - Указывает, сбрасывается ли EventWaitHandle автоматически или вручную после получения сигнала.

LazyThreadSafetyMode - Определяет, как экземпляр Lazy<T> синхронизирует доступ из нескольких потоков.

LockRecursionPolicy - Указывает, можно ли несколько раз войти в блокировку из одного и того же потока.

ThreadPriority - Задает приоритет выполнения потока Thread.

ThreadState - Задает состояния выполнения объекта Thread.

Делегаты

ContextCallback - Представляет метод, вызываемый в новом контексте.

IOCompletionCallback - Получает код ошибки, количество байтов и тип перекрывающегося значения при завершении операции ввода-вывода в пуле потоков.

ParameterizedThreadStart - Представляет метод, который выполняется в отношении Thread.

SendOrPostCallback - Указывает метод, вызываемый при отправке сообщения в контекст синхронизации.

ThreadExceptionEventHandler - Представляет метод, обрабатывающий событие ThreadExceptionApplication.

ThreadStart - Представляет метод, который выполняется в отношении Thread.

TimerCallback - Представляет метод, обрабатывающий вызовы от события Timer.

WaitCallback - Представляет метод обратного вызова, выполняющегося потоком из пула потоков.

WaitOrTimerCallback - Представляет метод, который вызывается при получении объектом WaitHandle сигнала или истечении времени ожидания.

Вывод

Я изучил пространство имен System.Threading, изучил как пользоваться потоками, в очередной раз осознал всю сложность и многогранность ООП на примере работы с потоками, и что нужно дальше изучать особенности языка C# такие как лямбда функции, делегаты, а также необходимость знаний системного программирования, поскольку System.Threading представляет собой работу с ресурсами компьютера.