2 слайд.

TPL.Dataflow -  это библиотека обработки данных, это часть .NET Framework, которая обеспечивает легкую и эффективную модель программирования для обработки данных в асинхронной, параллельной и потокобезопасной среде

Основой TPL.Dataflow являются блоки данных и связи между ними. Блоки данных представляют собой отдельные единицы обработки, которые могут принимать входные данные, обрабатывать их и генерировать выходные данные. Связи между блоками определяют направление потока данных и указывают, как данные передаются между блоками.

3 слайд

TPL определяет три вида блоков потока данных: исходные блоки, целевые блоки и блоки распространения. Исходный блок действует как источник данных и данные могут быть прочитаны из него. Целевой блок действует как приемник данных и данные могут быть записаны в него. Блок распространения действует как исходный блок, так и целевой блок, и данные могут быть прочитаны из него и записаны в него.

DataflowBlock – базовый класс блоков, необходим для реализации своих собственных, путём наследования.

ActionBlock – этот класс представляет блок данных, который выполняет какие-либо действия над входными данными без генерации выходных данных. Вы можете определить логику действия, реализуя делегат `Action<T>`, который принимает входные данные. Также делегат может быть System.Func<TInput, Task>.

TransformBlock – этот класс представляет блок данных, который выполняет преобразование входных данных и генерирует выходные данные. Вы можете определить логику преобразования, реализуя делегат `Func<TInput, TOutput>`, который принимает входные данные и возвращает выходные данные.

BroadcastBlock - этот класс представляет блок данных, который принимает входные данные и отправляет их на все связанные блоки. Это позволяет множественным блокам обрабатывать одни и те же данные параллельно.

JoinBlock – этот класс представляет блок данных, который принимает входные данные из нескольких источников и объединяет их в один набор выходных данных. Это позволяет объединять данные из разных источников для дальнейшей обработки.

На самом деле блоков больше, это лишь часть из них.

4 слайд

TPL.Dataflow автоматически обрабатывает данные параллельно, используя пул потоков.

TPL.Dataflow предоставляет возможность управлять пропускной способностью обработки данных, чтобы избежать перегрузки системы или недостатка ресурсов.

Блоки данных в TPL.Dataflow автоматически обрабатывают и передают ошибки вниз по графу. Это упрощает обработку ошибок.

TPL.Dataflow обеспечивает потокобезопасность при работе с блоками данных и передаче данных между ними, что позволяет безопасно использовать их в многопоточных сценариях.

5 слайд

На данном примере мы создаём три блока inputBlock, processBlock, outputBlock соответственно. В последующих строчках кода мы используем метод LinkTo. В качестве опции указываем PropagateCompletion = true, нужно для того, чтобы отслеживать их завершение (и ошибки). (Получает или задает значение, указывающее, будут ли уведомления о завершении и об ошибке автоматически распространятся на связанный целевой объект.)

.Complete() указывает, что данные больше не будут приходить. Получает объект [Task](https://learn.microsoft.com/ru-ru/dotnet/api/system.threading.tasks.task?view=net-5.0), представляющий асинхронную операцию и завершение блока потока данных.

7 слайд

Все это нужно для следующих проектов

* Проекты с высокой нагрузкой
* Проекты с параллельной обработкой данных
* Проекты, где необходима обработка ошибок и отказоустойчивость

8 слайд

TPL.Dataflow – мощный инструмент для обработки данных, обладающий гибкостью и предлагающий различные варианты для создания конвейеров обработки данных.