# ЛАБ15

1. Что такое tpl? Как и для чего используется тип task

Tpl (task parallel library) - это библиотека в .net framework, предоставляющая удобные средства для параллельного программирования. Она включает в себя тип task, который представляет асинхронную операцию или исполнение кода в фоновом потоке.

1. Почему эффект от распараллеливания наблюдается на большом  
   количестве элементов?

Эффект от распараллеливания наблюдается на большом количестве элементов, потому что при параллельной обработке каждый элемент может быть обработан независимо от других элементов. Параллельное выполнение позволяет использовать множество ядер и потоков процессора для одновременной обработки данных, ускоряя выполнение задачи.

1. В чем основные достоинства работы с задачами по сравнению с  
   потокми?

Простота использования, автоматическое управление потоками, улучшенная масштабируемость, возможность работы с результатами и исключениями.

1. Приведите три способа создания и/или запуска task?

Task + Start()

Task + Run()

Task+Factory+StartNew()

1. Как и для чего используют методы wait(), waitall() и waitany()?

Для ожидания завершения задач.

* Wait() блокирует текущий поток до завершения задачи.
* WaitAll() блокирует текущий поток до завершения всех задач из указанного массива задач.
* WaitAny() блокирует текущий поток до завершения любой из задач из указанного массива задач.

1. Приведите пример синхронного запуска task?

|  |
| --- |
| Task task = Task.Run(() => {  ….  });  task.Wait(); |

1. Как создать задачу с возвратом результата?

Использовать обобщенный тип Task<TResult>

1. Как обработать исключение, если оно произошло при выполнении  
   task?

Для обработки исключений, возникающих при выполнении задачи, можно использовать свойство Exception задачи или метод ContinueWith() для добавления задачи продолжения, которая будет выполняться только в случае возникновения исключения.

1. Что такое cancellationtoken и как с его помощью отменить  
   выполнение задач?

CancellationToken - это механизм для отмены выполнения задачи или операции. Он используется для передачи сигнала отмены из одного потока в другой.

1. Как организовать задачу продолжения (continuation task) ?

Задачу продолжения (continuation task) можно организовать с помощью метода ContinueWith().

1. Как и для чего используется объект ожидания при создании задач  
   продолжения?

Объект ожидания (Task) используется при создании задач продолжения для управления потоком выполнения.

1. Поясните назначение класса system.threading.tasks.parallel?  
   Класс System.Threading.Tasks.Parallel предоставляет статические методы для выполнения параллельных операций.
2. Приведите пример задачи с parallel.for(int, int, action<int>)

|  |
| --- |
| Parallel.For(0, 10, i =>  {  Console.WriteLine($"Task {i} is running on thread  {  Thread.CurrentThread.ManagedThreadId}");  }); |

1. Приведите пример задачи с parallel.foreach

|  |
| --- |
| string[] names = { "Alice", "Bob", "Charlie" };  Parallel.ForEach(names, name =>  {  Console.WriteLine($"Hello, {name}! Thread:  {  Thread.CurrentThread.ManagedThreadId}");  }); |

1. Приведите пример с parallel.invoke()

|  |
| --- |
| Parallel.Invoke(  () => DoWork("Task 1"),  () => DoWork("Task 2"),  () => DoWork("Task 3")  ); |

1. Как с использованием cancellationtoken отменить параллельные  
   операции?

Для отмены параллельных операций с использованием CancellationToken можно передать токен отмены при создании задачи или использовать его внутри задачи для проверки состояния отмены и прерывания выполнения.

1. Для чего используют blockingcollection<t>, в чем ее особенность?

BlockingCollection<T> используется в параллельном программировании для безопасной обработки данных между производителями и потребителями в многопоточной среде. Основная особенность BlockingCollection<T> заключается в том, что она предоставляет блокирующие операции добавления и извлечения элементов из коллекции, что позволяет потокам ожидать доступных данных или места для вставки.

1. Как используя async и await организовать асинхронное выполенение  
   метода?

Объявляется метод с ключевым словом async. Внутри метода используется ключевое слово await перед вызовом асинхронной операции.