Module 09:
"Concurrent Collections"

TEKNOLOGISK
INSTITUT

Agenda

- Introducing Concurrent Collections
- Examples of Usage

Task Parallel Library

- Task Parallel Library (TPL)
  - Was introduced in .NET 4.0
  - Enhanced in .NET 4.5
    - Special keywords are included in C# 5.0, C# 8.0

- Features
  - Task Parallelism
  - Data Parallelism
  - Parallel LINQ
  - Thread-safe collections

- Emerging trends leverage parallelism! Also .NET!

Introducing Concurrent Collections

- Task Parallel Library includes thread-safe collection alternatives to the conventional generic collections

- `System.Collections.Concurrent` namespace
  - `ConcurrentQueue<T>`
  - `ConcurrentStack<T>`
  - `ConcurrentDictionary<K,V>`
  - `ConcurrentBag<T>`

Agenda

- Introducing Concurrent Collections
- **Examples of Usage**

ConcurrentQueue<T>

- **ConcurrentQueue<T>** is a thread-safe version of **Queue<T>**

```
ConcurrentQueue<int> queue = new ConcurrentQueue<int>();

Task producer = Task.Factory.StartNew( () => { ...
   queue.Enqueue( DateTime.Now.Milliseconds );
   ...
}

Task consumer = Task.Factory.StartNew( () => { ...
   int number;
   if( queue.TryDequeue( out number ) ) { ... }
}
```

BlockingCollection<T>

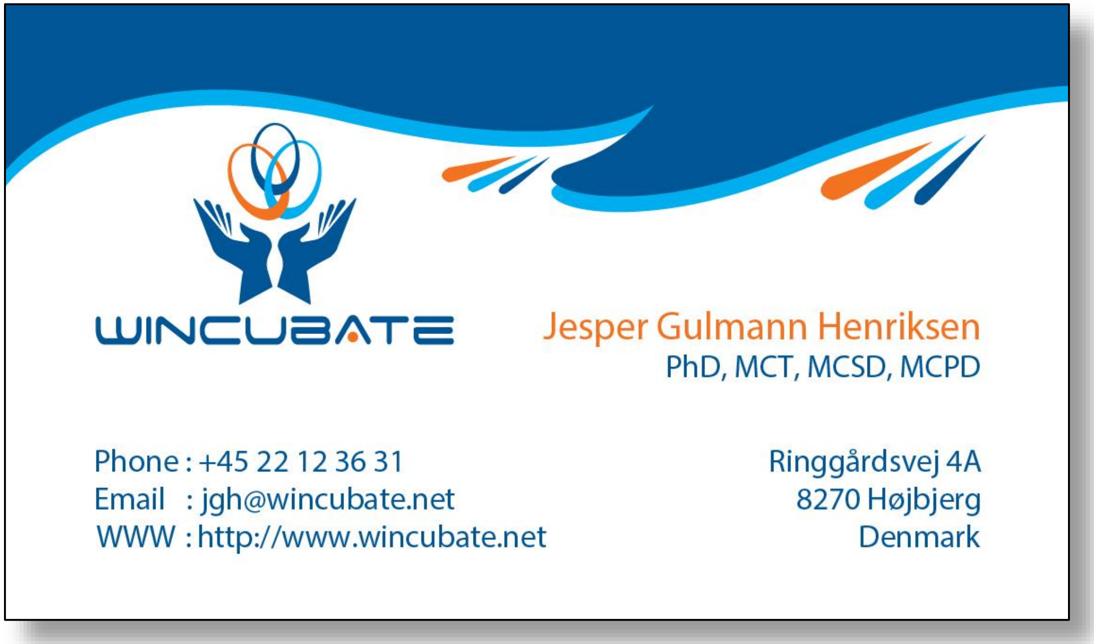
- `BlockingCollection<T>`
  - Concurrent collection
  - Optional bounded capacity
  - Blocking operations

```
BlockingCollection<int> bc = new BlockingCollection<int>( 5 );
...
string result = string.Format( $"Successfully took {0}",
   await Task.Run<int>( () => bc.Take() );
```

- Implement your own concurrent collection using
  - `IProducerConsumerCollection<T>`

Summary

- Introducing Concurrent Collections
- Examples of Usage



WINCUBATE

Jesper Gulmann Henriksen
PhD, MCT, MCSD, MCPD

Phone : +45 22 12 36 31
Email : jgh@wincubate.net
WWW : http://www.wincubate.net

Ringgårdsvej 4A
8270 Højbjerg
Denmark