ConcurrentDictionary Demo

Simon Robinson

@TechieSimon | TechieSimon.com

Module 3 Overview

- App based on Geek Clothing Store
- `AddOrUpdate()` lets us update stock levels no matter what other threads are doing
- Need to be careful how we write lambdas that may execute multiple times
- You can write thread-safe code without any explicit locks!

CODE DEMO

This slide must not appear in the recorded course

Avoiding Locks

CODE DEMO

Grey area must not appear in the recorded course

`BuyStock()` has no explicit locks!

✓ Likely to be scalable

⚠ But not atomic

CODE DEMO

This slide must not appear in the recorded course

CODE DEMO

Grey area must not appear in the recorded course

Thread-safe Conditional Updates

Put logic in delegate:

- ✓ It works!
- ✓ Efficient
- ✓ Often the best solution
- ⚠ Code can be complex

CODE DEMO

This slide must not appear in the recorded course

CODE DEMO

Grey area must not appear in the recorded course

Thread-safe Conditional Updates

Break into Simpler Operations:

✓ Code can be simpler

But in this case…

⚠ Stock levels go negative

⚠ Performance may be worse

CODE DEMO

Grey area must not appear in the recorded course

Thread-safe Conditional Updates

Lock threads:

```
lock (_syncObj)
{
    if (_stock[item] == 0)
        return false;
    _stock[item]--;
    _totalQuantitySold++;
    return true;
}
```

✓ One-method rule doesn't apply

⚠ Scalability

✗ Often not a good solution

CODE DEMO

This slide must not appear in the recorded course

Module 3 Summary

- Demo with multiple threads updating a `ConcurrentDictionary`
  - One concurrent dictionary method call for each operation
  - We worked around this by breaking an operation into multiple simpler operations
  - Delegates to `AddOrUpdate()` and `GetOrAdd()` can execute multiple times – so beware of side-effects