

# Introduction to Reactive Extensions

How Miranda racted the *Young Lady's Illustrated Primer*



# Overview

- Extends LINQ / .NET
- Observable sequence
- Observer pattern
- Concurrency
- Completion and Errors

# Overview

- ➔ Extends LINQ / .NET
  - Observable sequence
  - Observer pattern
  - Concurrency
  - Completion and Errors

# Rx

- System.Reactive
- System.Reactive.dll
- References
  - { } -
  - { } System
  - { } System.Collections.Generic
  - { } System.Joins
  - { } **System.Linq**
    - GroupedObservable<TKey,TElement>
    - IGroupedObservable<TKey,TElement>
    - Notification
    - Observable
    - ObservableAwaiter<TSource>
    - ObservableQuery
    - ObservableQuery<TSource>
    - ObservableQueryProvider
    - Observer
    - Qbservable

# Overview

- ➔ Extends LINQ / .NET
  - Observable sequence
  - Observer pattern
  - Concurrency
  - Completion and Errors

```
var nullField = from d in directories
                 where f.name = d.name
                 select f d;
```

# Overview

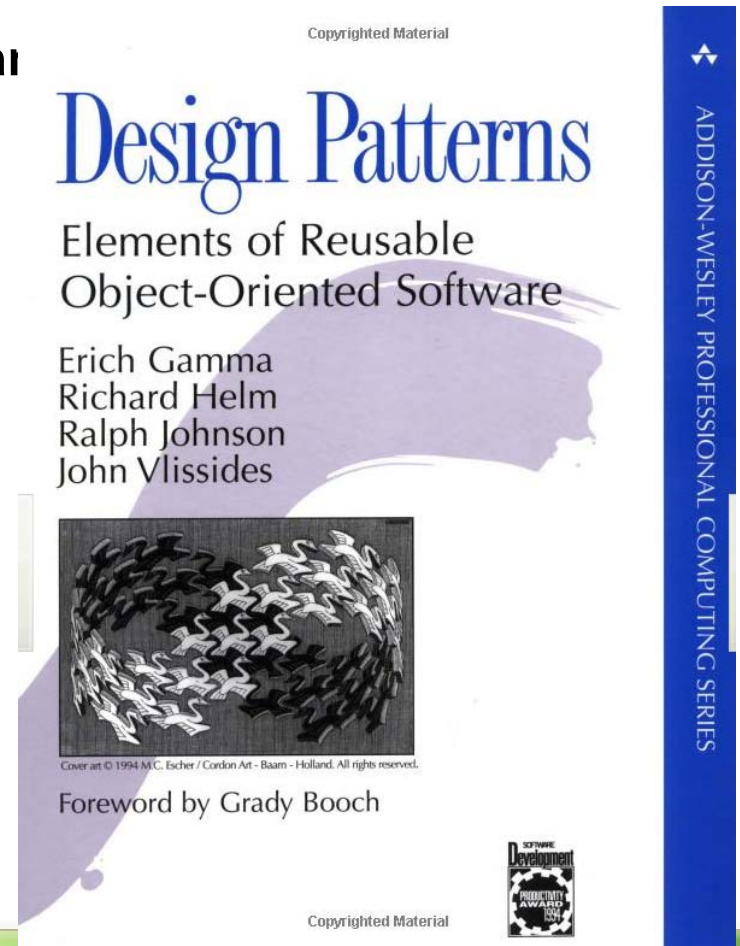
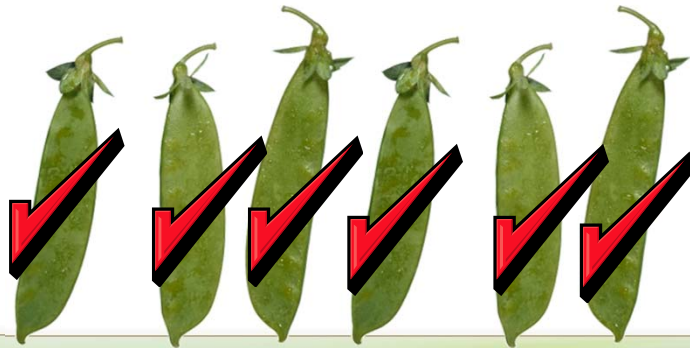
- Extends LINQ / .NET
- Observable sequence
- Observer pattern
- Concurrency
- Completion and Errors

```
observable.Subscribe(ProcessMe, ImDone);
```



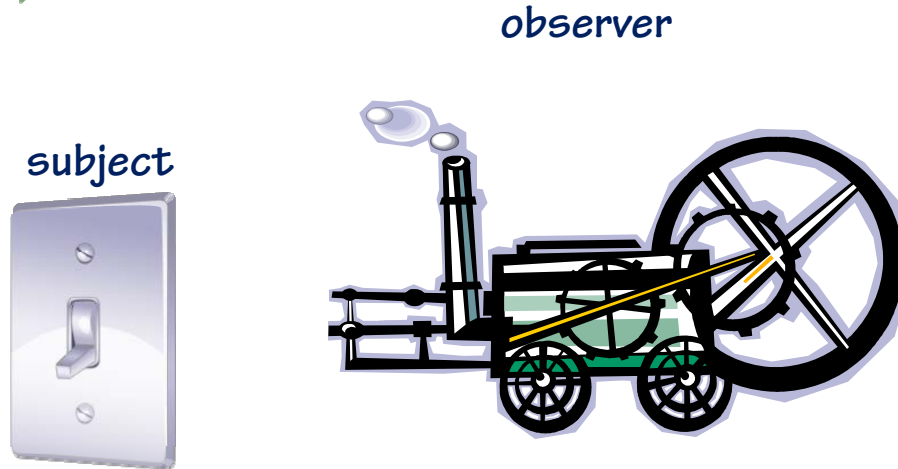
# Observer Pattern

- GOF
- ➔ General design description with solution at hand
- Light observes switch



# Observer Pattern

- GOF
  - General design description with solution and consequences
- ➔ Light observes switch



```
class Bulb
{
    SetFilament(On|Off)
}
```

```
class Switch
{
    SetLever (Up|Down)
    { SetFilament(..) }
}
```

# Observer Pattern

- GOF
- General design description with solution and consequences

➔ Light observes switch

```
class abstract Subject
{
    abstract Attach(Observer);
    abstract void Detach(Observer);
    abstract void Notify();
}
```

```
class abstract Observer
{
    abstract void Update();
}
```



# Observer Pattern

- GOF
- General design description with solution and consequences

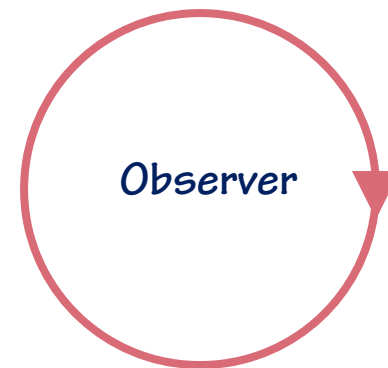
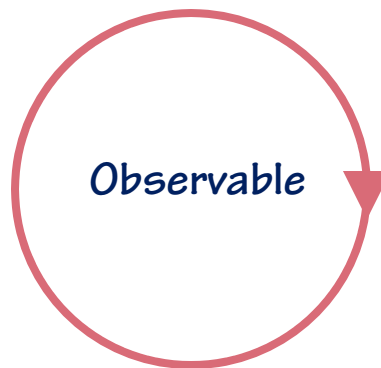
➡ Light observes switch

```
interface IObservable
{
    Subscribe();
}
```

```
interface IObservable
{
    OnNext();
    OnCompleted();
    OnError();
}
```

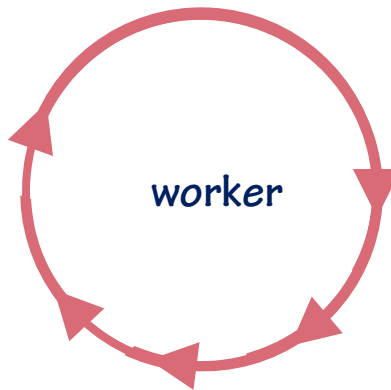
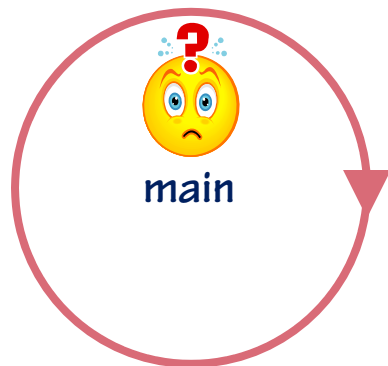
# Concurrency

- Observable thread controlled by SubscribeOn
- Observer thread controlled by ObserveOn
- Number of thread choices including Dispatcher



# Completion and Errors

- ➔ **Concurrency introduces two issues**
  - Are we there yet?
  - Oops!
- ➔ **Completion/Error is on "other" thread**



# Summary

- ➔ **Reactive Extensions implement the Observer Pattern**
- ➔ **Convert LINQ queries into observable sequences**
  - IObserver
  - IObservable
- ➔ **Manages concurrency**
  - query
  - observer

# References

- **Reactive Extensions Home**
  - <http://msdn.microsoft.com/en-us/data/gg577609>
- ***The Diamond Age***
  - Neal Stephenson ISBN 0553380966
- **Design Patterns: Elements of Reusable Object-Oriented Software**
  - *Gamma et al.* ISBN 0201633612
- **Pluralsight Design Pattern Library**
  - <http://www.pluralsight-training.net/microsoft/olt/Course/Toc.aspx?n=patterns-library>
- **ReSharper**
  - <http://www.jetbrains.com/resharper/>