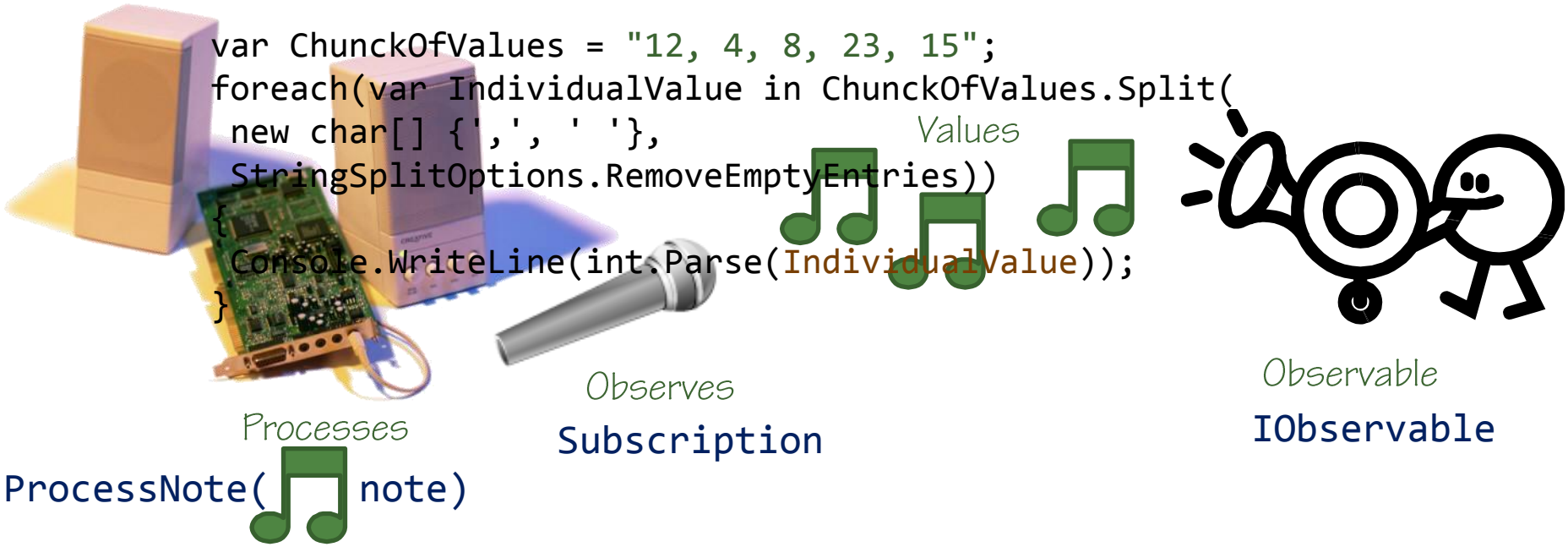
Observables

"Call me if you find out anything."

Overview

- Observable
- Observations ≡ Callbacks
- Generating Callbacks



Observable Sequence

- Subscribe
- Grammar

IObservable

starts callbacks

creates observable

```
var observable.Subscribe(Console.WriteLine, HandleError, Done)
```

IObservable

```
IDisposable Subscribe(IObserver observer)
{
   foreach(var i in new int[] {1,2,3})
   {
      observer.OnNext(i);
   }
   return this;
}
```

OnNext 1;
OnNext 2;
OnNext 3;

OnNext* (OnError | OnComplete)?

Create

- **Implement IObserver.Subscribe**
  - Ad hoc observable sequence
  - No dependence on LINQ query or IEnumerable
- **Observable.Create Observable.CreateWithDisposable**

```
IEnumerable<int> MyAdHocSequence()
{
        yield return 1;
        yield return 2;
}

var observable = MyAdHocSequence().ToObservable();
```

Run

- **Runs subscription**
  - blocks until complete

Start

- **Subscribes to observable**
  - returns list
  - no OnNext, OnError, or OnComplete
- **Invokes a delegate**
  - IObservable<Type>
  - IObservable<System.Unit>

Summary

- ObservableExtensions.Subscribe creates IObservable.Subscribe
- OnNext* (OnError | On Complete)
- Create for ad hoc Subscribe
- Run to wait for completion
- Start
  - ListObservable
  - delegate

```
IObservable.Subscribe(IObserver observer)
{
   observer.OnNext(<value>);
   observer.OnNext(<value>);
   observer.OnCompleted();
}
```

References

- **Redgate Reflector**
  - http://www.reflector.net/