

C# Generics

Generic Constraints, Contravariance, Covariance

K. Scott Allen
@OdeToCode



pluralsight
hardcore developer training

Generics So Far...

```
public virtual void Write(T value)
{
    _queue.Enqueue(value);
}

public virtual T Read()
{
    return _queue.Dequeue();
}
```

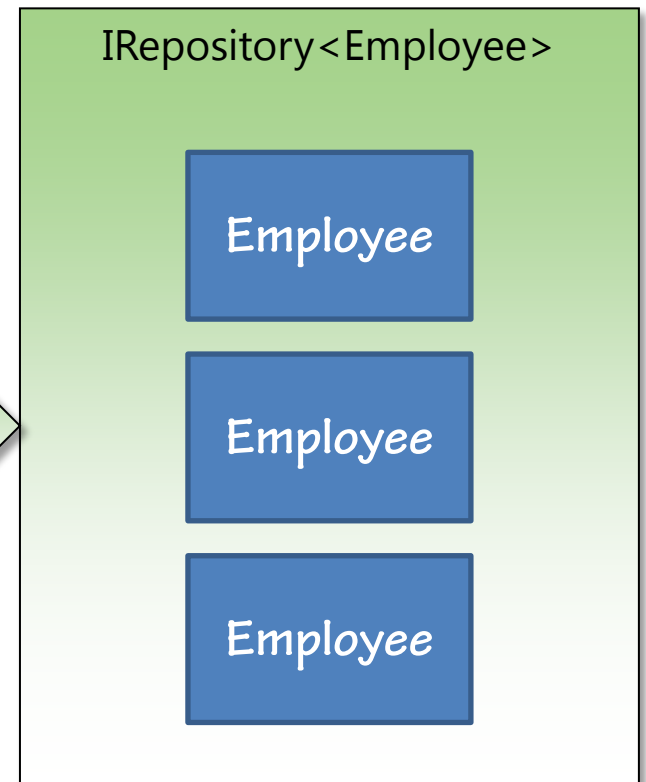
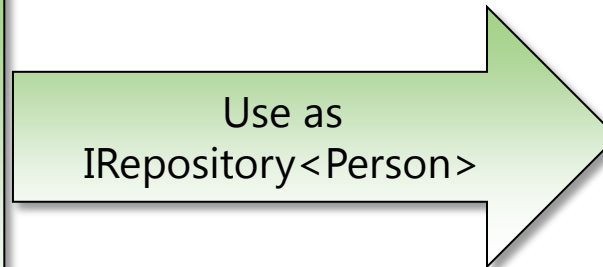
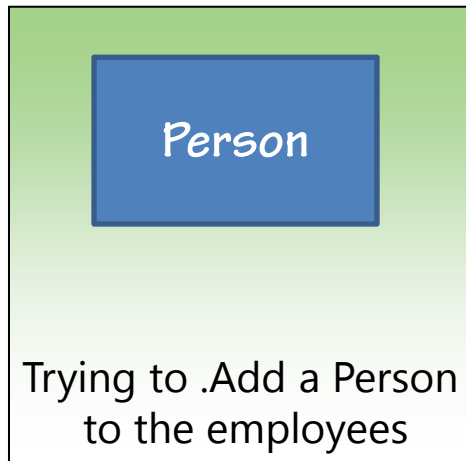
The Problem

```
if (value.IsValid())  
{  
    _queue.Enqueue(value);  
}
```

Constraints

- **Force a type parameter to have certain characteristics**
 - Be a reference type or value type
 - Implement an interface
 - Derive from a base class
 - Have a default constructor
 - Be an instance derive from another generic type parameter

```
IBuffer<T> where T: class, IItem, new()
```



```
public class EntityFinder<TContext> where TContext : DbContext
{
    public TEntity Find<TEntity>(int id) where TEntity : class
    {
        return null;
    }
}
```

```
delegate TOutput Converter<in TInput, out TOutput>(TInput input);
```

