

# Contract-based Software Development

Rasmus Guldborg Pedersen

January 2015

# Overview

## 1 Class Specification

# Interface

```
public interface ISimpleQueue {  
    void Enqueue(object item);  
    object Dequeue();  
    object ElementAt(int index);  
    int Count();  
}
```

# Contract

```
abstract class ISimpleQueueContract {  
    public void Enqueue(object item) {  
        Contract.Requires(item != null);  
        Contract.Ensures(Count ==  
            Contract.OldValue(Count()) + 1);  
        Contract.Ensures(ElementAt(Count()) == item);  
        // ...  
    }  
    // ...  
}
```

# Associating Interface with Contract

```
[ContractClass(typeof(ISimpleQueueContract))]  
public interface ISimpleQueue { /* ... */ }  
  
[ContractClassFor(typeof(ISimpleQueue))]  
abstract class ISimpleQueueContract { /* ... */ }
```

# The End

*“Testing shows the presence, not the absence of bugs.”  
— Edsger W. Dijkstra*