

# Contract-based Software Development

Rasmus Guldberg Pedersen

January 2015

# Overview

- 1 Invariants
- 2 Code Contracts
  - Overview
  - Object Invariants

# Object invariant

What is an object invariant? What is Code Contracts? How can it be used to specify a class invariant?

# Invariants

Property of “something” that doesn't change.

# Object Invariants

- Observable property
- Doesn't change
- + conceptual model

# Code Contracts (.NET tool)

Express preconditions, postconditions and object invariants for:

# Code Contracts (.NET tool)

Express preconditions, postconditions and object invariants for:

- Static analysis

# Code Contracts (.NET tool)

Express preconditions, postconditions and object invariants for:

- Static analysis
- Documentation



# Code Contracts (.NET tool)

Express preconditions, postconditions and object invariants for:

- Static analysis
- Documentation
- Runtime checking

# Simple Example

```
[ContractInvariantMethod]  
private void ObjectInvariant() {  
    Contract.Invariant( this .y >= 0 );  
    Contract.Invariant( this .x > this.y );  
    ...  
}
```

# Usefull Object Invariant

```
public class BoundedQueue {  
    public BoundedQueue(int maxSize) { /* ... */ }  
    public void Enqueue(object item) {  
        Contract.Ensures(Count ==  
            Contract.OldValue(Count) + 1); // ...  
    }  
    public object Dequeue() {  
        Contract.Requires(Count > 0);  
        Contract.Ensures(Count ==  
            Contract.OldValue(Count) - 1); // ...  
    }  
    public int Count { get; private set; }  
    public int MaxCount { get; private set; }  
}
```

# Usefull Object Invariant

```
[ContractInvariantMethod]
private void ObjectInvariant(){
    Contract.Invariant(Count >= 0);
    Contract.Invariant(Count <= MaxCount);
    Contract.Invariant(MaxCount ==
        Contract.OldValue(MaxCount));
}
```

# The End

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