

# C# Interfaces

---

## INTRODUCING INTERFACES



**Jeremy Clark**

DEVELOPER BETTERER

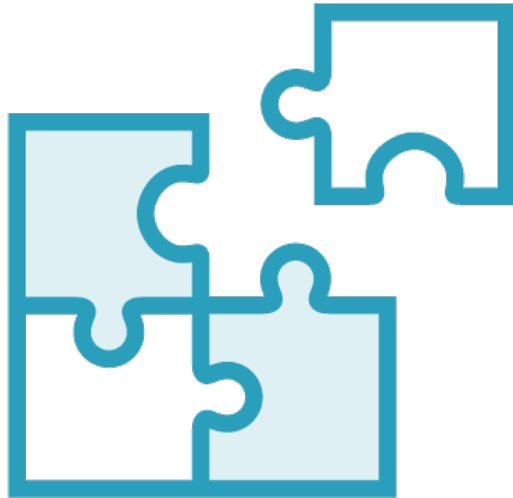
@jeremybytes [www.jeremybytes.com](http://www.jeremybytes.com)



# Why Interfaces?



Maintain



Extend



Test

# Trying to Learn Interfaces



Interfaces?



IFoo and bar class



NO!



Much reading



Conversations



This is awesome!



# Our Roadmap

**What?**

**Definition and technical bits**

**Why?**

**Maintain, extend, test**

**How?**

**Create and implement**

**Where?**

**Practical bits, dependency  
injection, and mocking**



# Prerequisites

**Classes**

**Inheritance**

**Properties**

**Methods**



# What & Why



## Definition

## Differences

Concrete classes

Abstract classes

Interfaces

## Interfaces and flexible code



Interfaces describe a group of related functions that can belong to any class or struct.





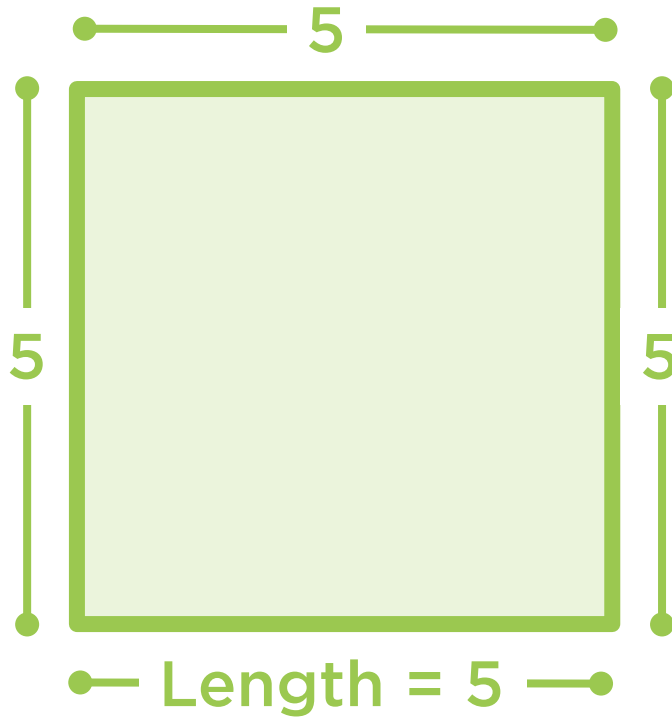
## Contract

“I have these functions.”

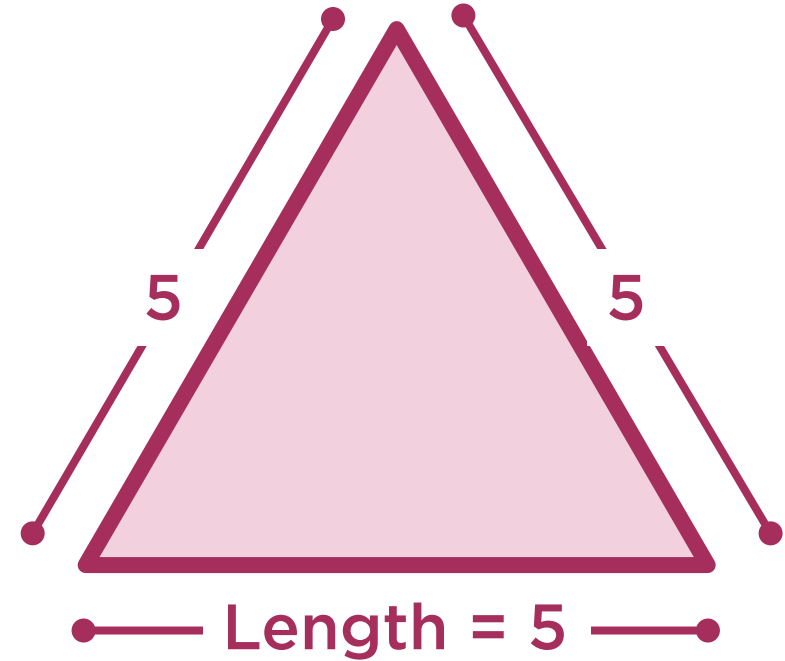
Properties, methods, events, indexers



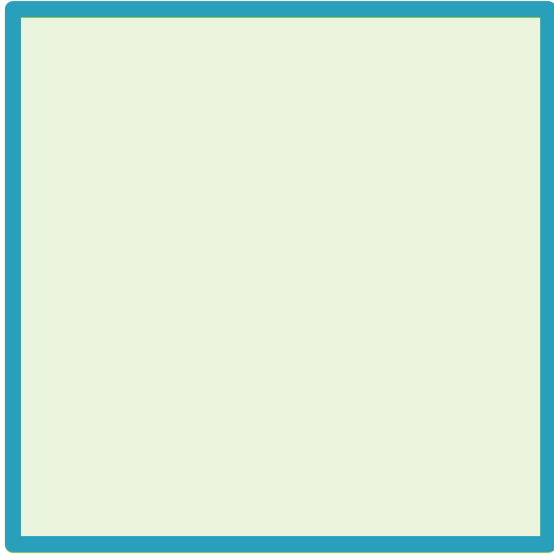
# Regular Polygons



3 or more sides  
Sides are same length

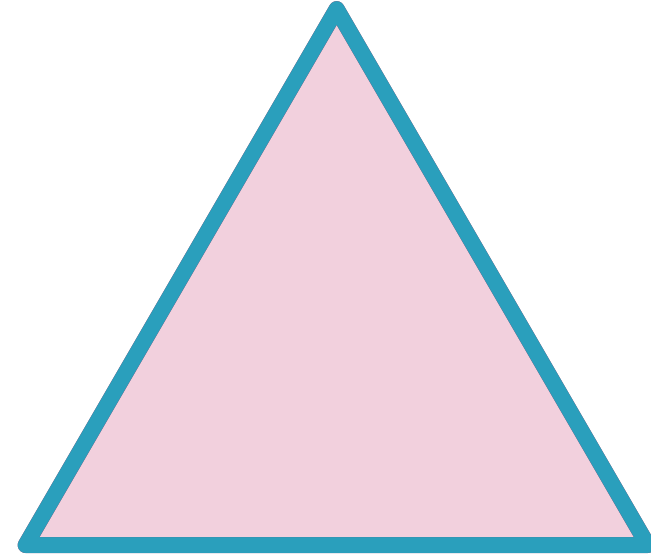


# Perimeter



Length = 5

perimeter = sides x length



Length = 5

perimeter = sides x length

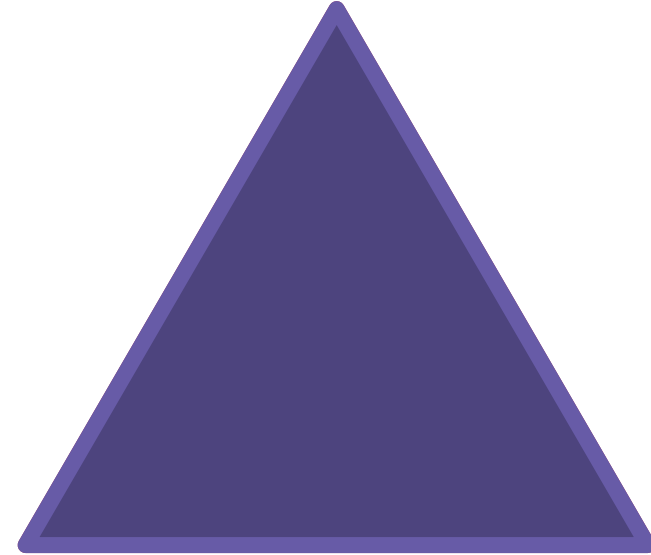


# Area



Length = 5

area = length x length



Length = 5

area = length x length x sqrt(3) / 4



# Demo



## Regular polygon as

- Concrete class
- Abstract class
- Interface



# Interfaces and Flexible Code

**Resilience in the  
face of change**

**Insulation from  
implementation details**





Program to an abstraction rather  
than a concrete type



Program to an interface rather  
than a concrete class



## Concrete Collection Types

List<T>

Array

ArrayList

SortedList<TKey, TValue>

HashTable

Queue / Queue<T>

Stack / Stack<T>

Dictionary<TKey, TValue>

ObservableCollection<T>

◀ Collection with add/remove

◀ Unordered bag of objects

◀ First in / first out collection

◀ Last in / first out collection





## List<T> Interfaces

```
public class List<T> :  
    IList<T>, IList,  
    ICollection<T>,  
    IReadOnlyList<T>,  
    IReadOnlyCollection<T>,  
    IEnumerable<T>, IEnumerable
```

◀ Allows iteration

Used with  
foreach  
List boxes  
LINQ



# Demo



Code against a class and an interface

Change the type

See how the code responds



# What & Why



## Definition

## Differences

Concrete classes

Abstract classes

Interfaces

## Interfaces and flexible code

