Understanding Interfaces

Deborah Kurata
http://msmvps.com/blogs/deborahk/
@DeborahKurata
deborahk@insteptech.com

pluralsight
hardcore dev and IT training

Interface

User Interface

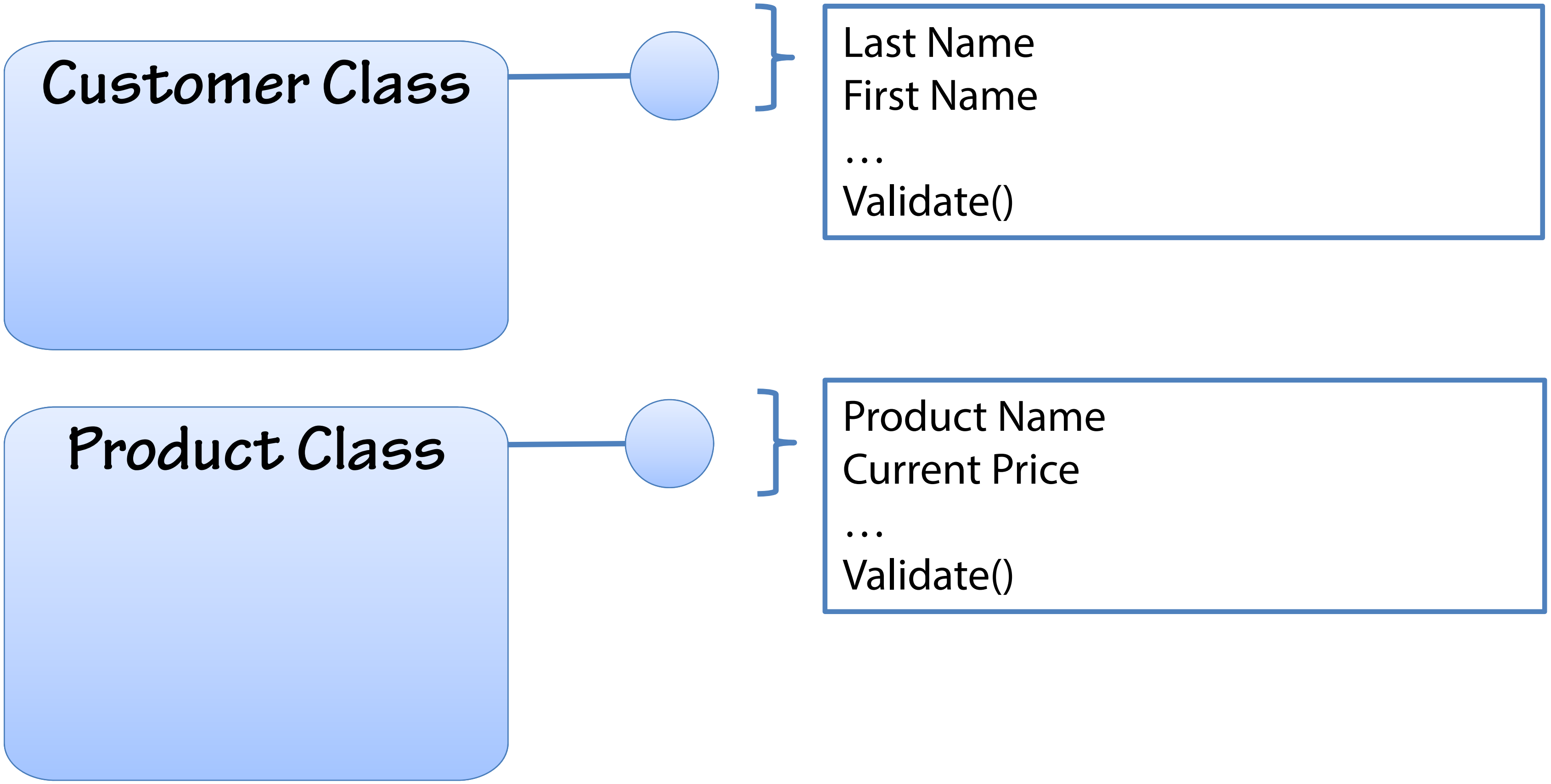
Web API

In computer science, an **interface** is a shared boundary across which two separate components of a computer system exchange information. The exchange can be between software, computer hardware, peripheral devices, humans and combinations of these.

- Wikipedia 5/3/14

Class Interface

Implicit Class Interface



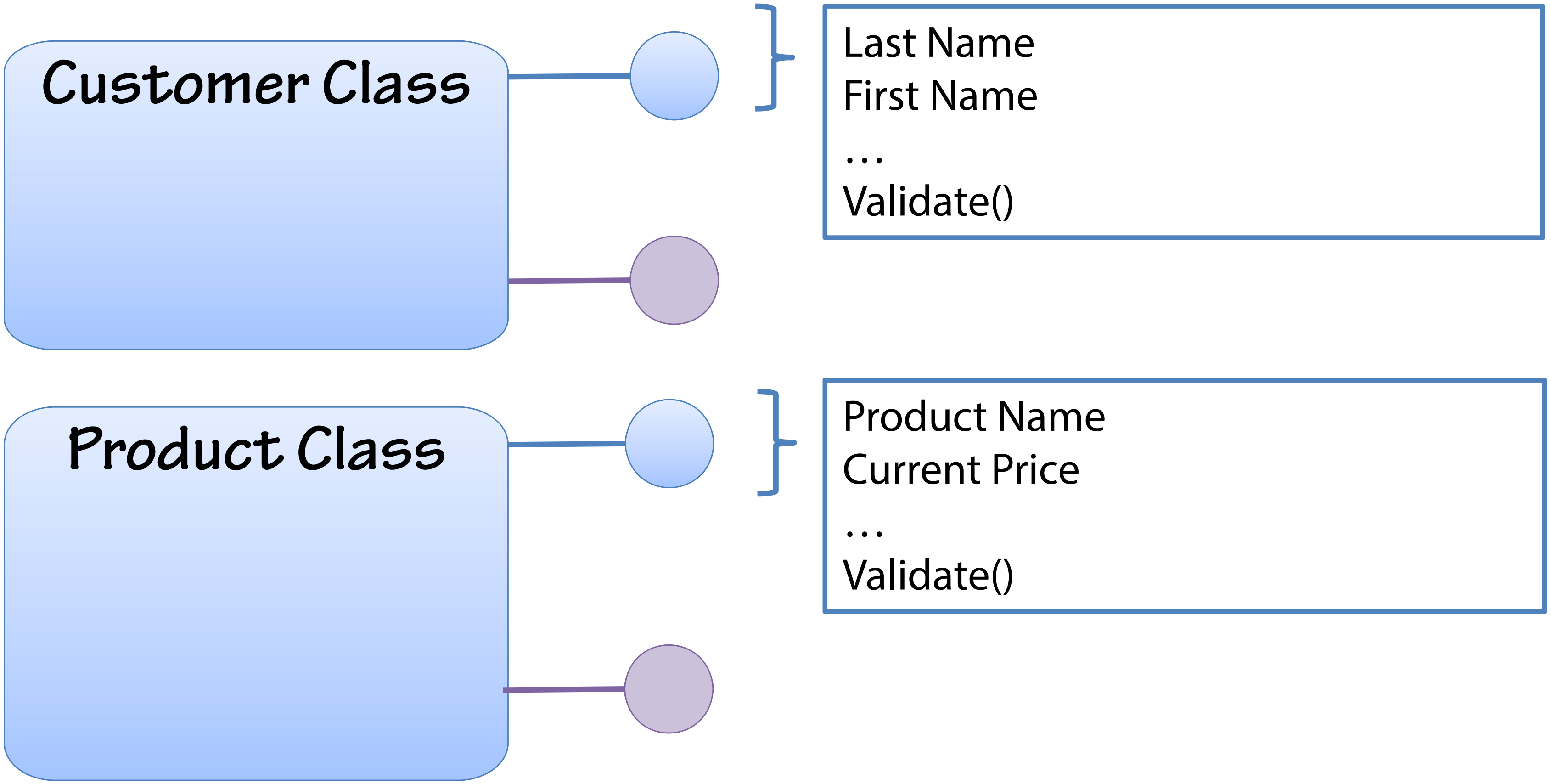
Customer Class

Last Name
First Name
…
Validate()

Product Class

Product Name
Current Price
…
Validate()

Explicit Interfaces



Customer Class

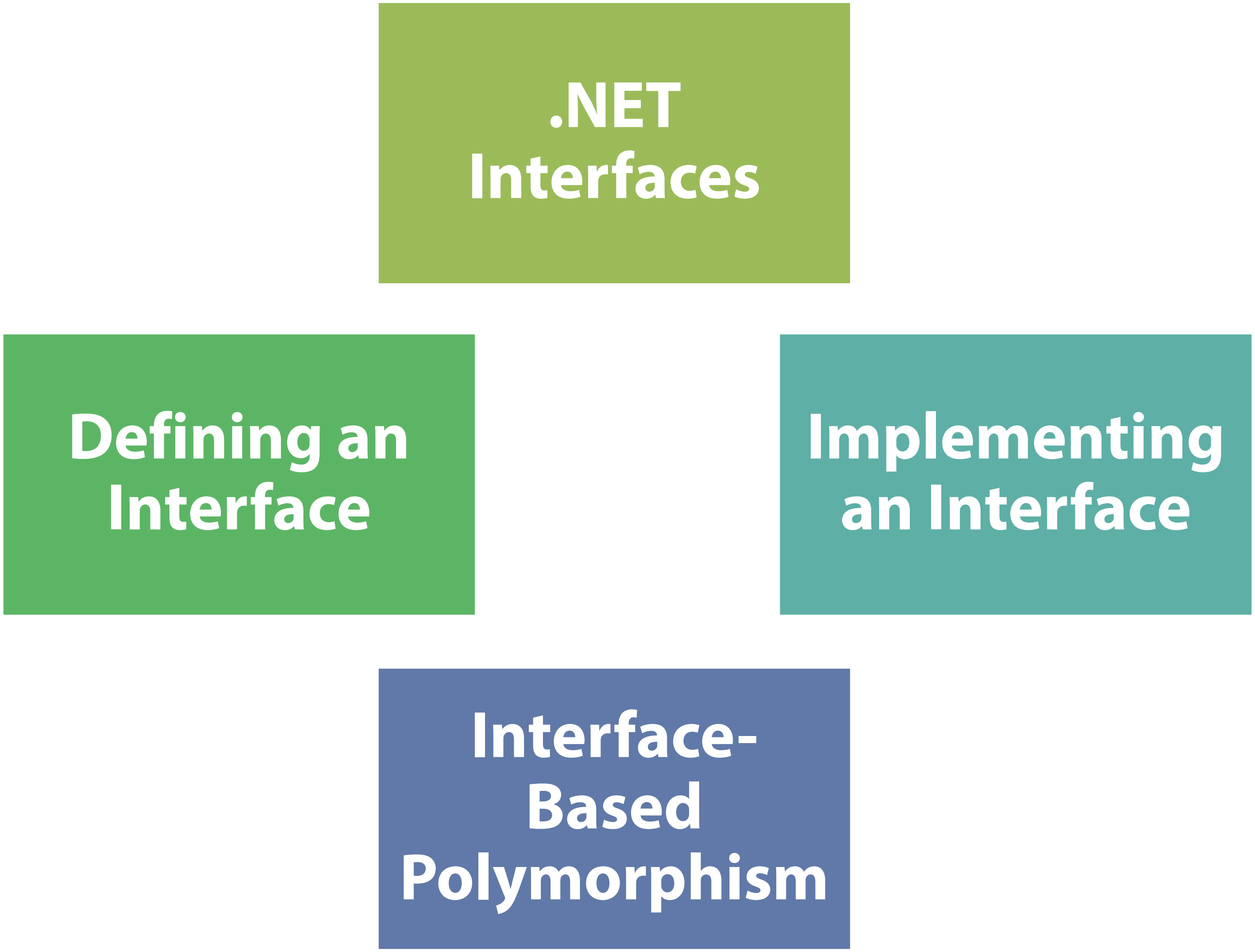
Last Name
First Name
…
Validate()

Product Class

Product Name
Current Price
…
Validate()

```
public interface ILoggable
string Log()
```

Module Outline



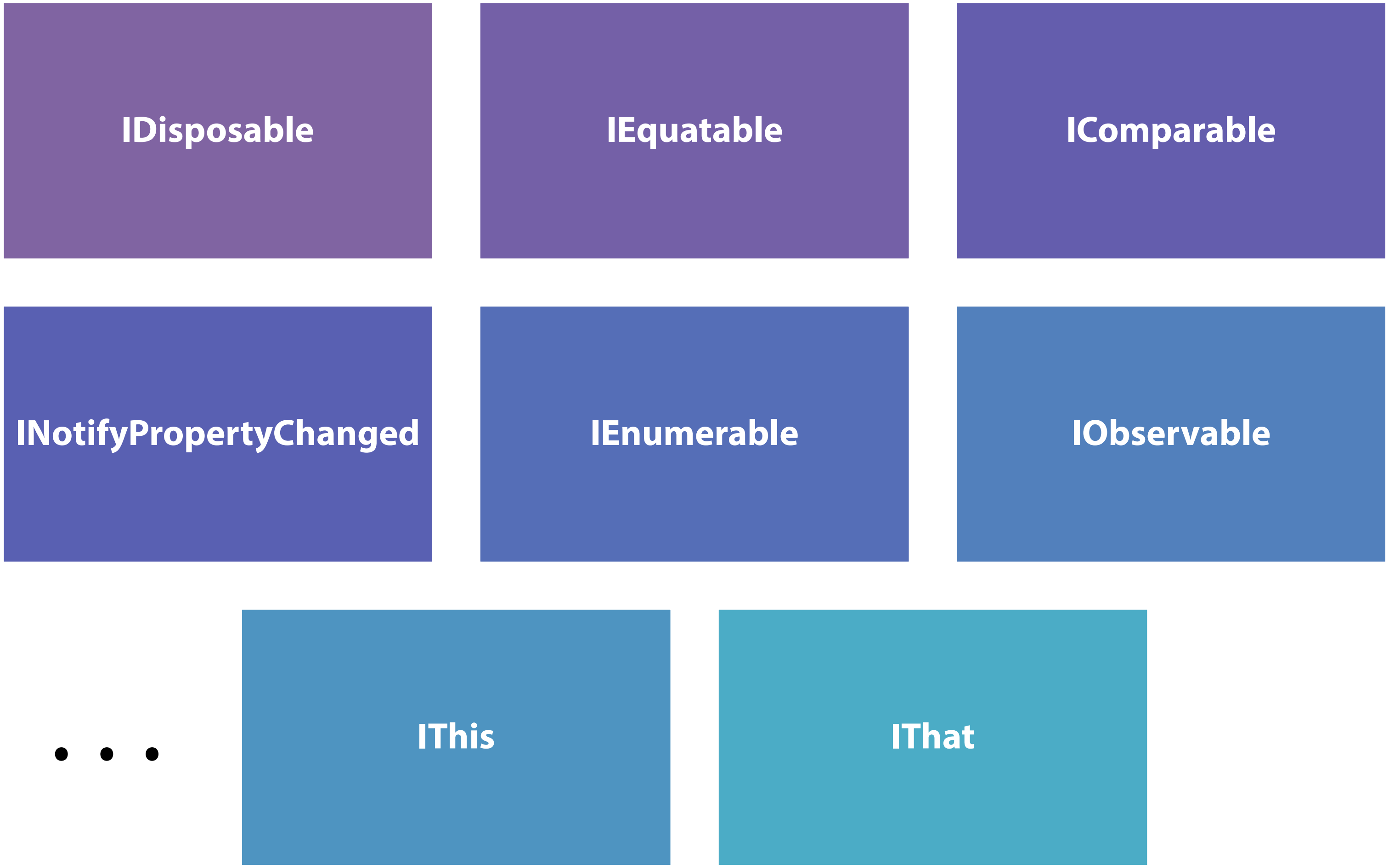
.NET Interfaces

Defining an Interface

Implementing an Interface

Interface-Based Polymorphism

.NET Framework Interfaces



IDisposable

IEquatable

IComparable

INotifyPropertyChanged

IEnumerable

IObservable

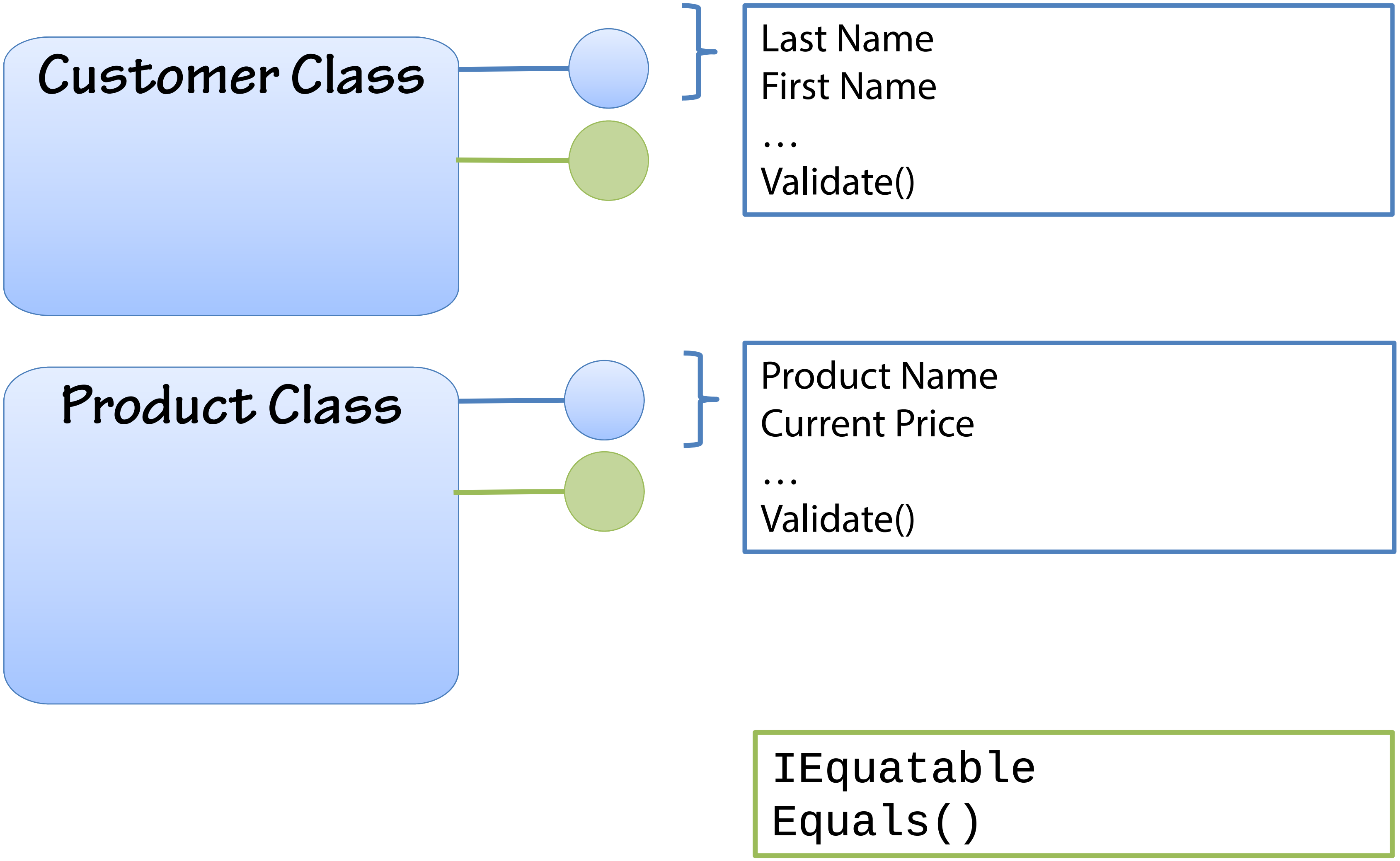
. . .

IThis

IThat

```
public void RetrieveExisting()
       {
        //-- Arrange
        var customerRepository = new CustomerRepository();
        var expected = new Customer(1)
        {
            EmailAddress = "fbaggins@hobbiton.me",
            FirstName = "Frodo",
            LastName = "Baggins"
        };

        //-- Act
        var actual = customerRepository.Retrieve(1);

        //-- Assert
        //Assert.AreEqual(expected, actual);

        Assert.AreEqual(expected.CustomerId, actual.CustomerId);
        Assert.AreEqual(expected.EmailAddress, actual.EmailAddress);
        Assert.AreEqual(expected.FirstName, actual.FirstName);
        Assert.AreEqual(expected.LastName, actual.LastName);
    }
```

.NET IEquatable Interface



Customer Class

Last Name
First Name
…
Validate()

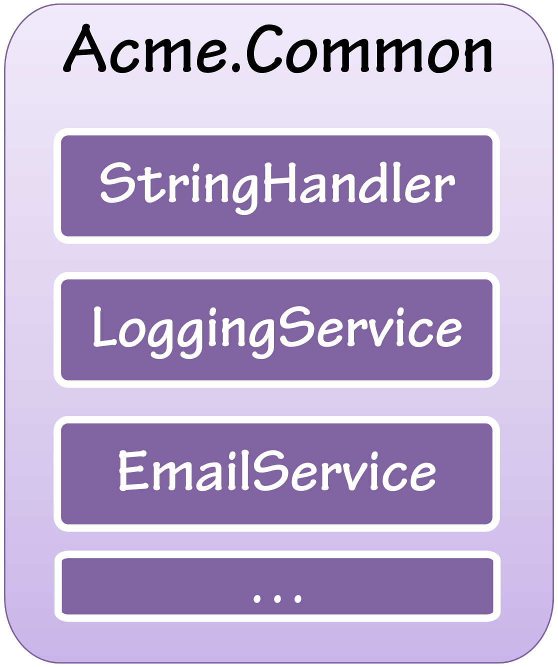
Product Class

Product Name
Current Price
…
Validate()

```
IEquatable
Equals()
```

Logging

- Resolving Bugs
- Security
- Data Analysis


Acme.Common

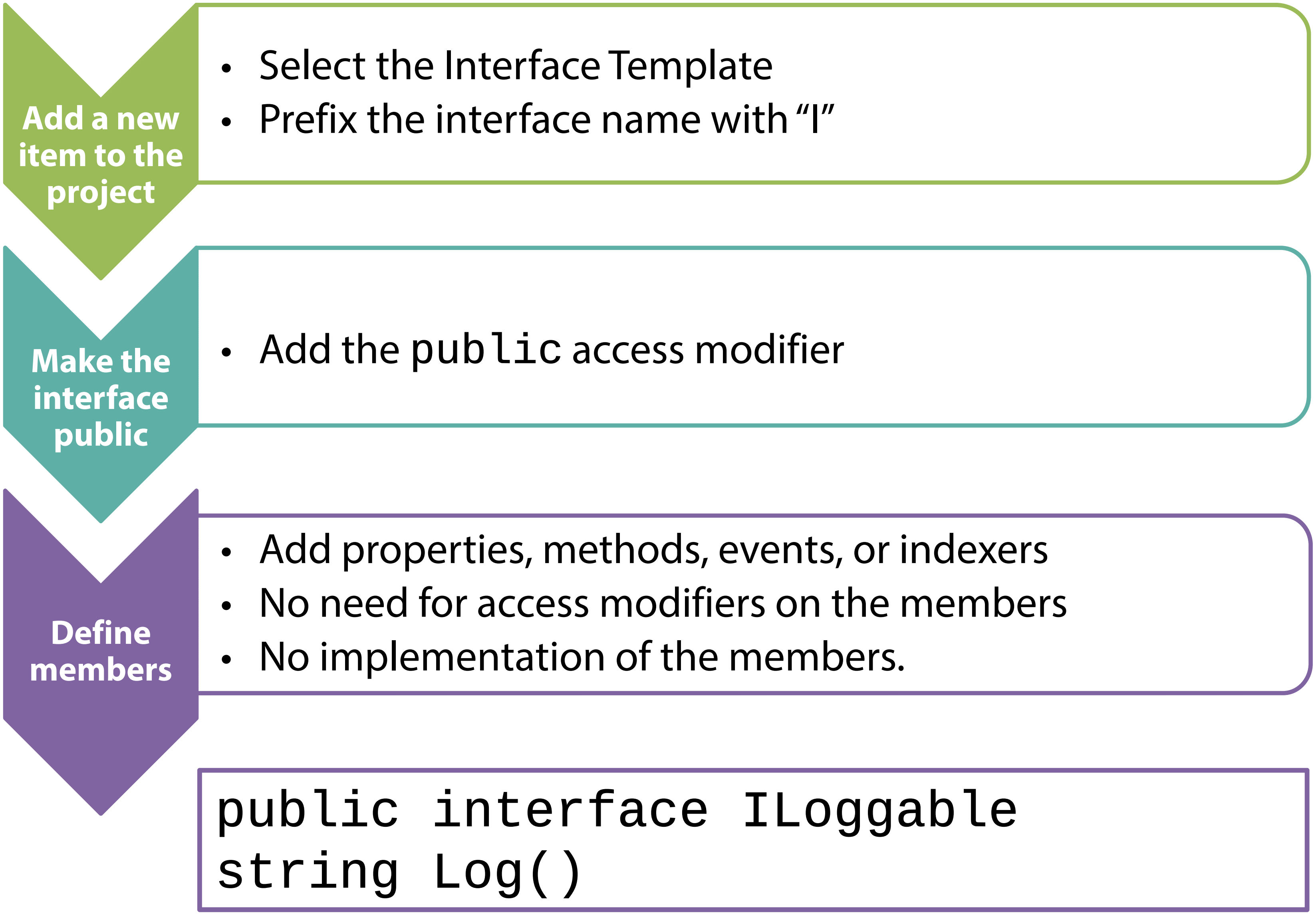
StringHandler

LoggingService

EmailService

...

Defining an Interface



Add a new item to the project

- Select the Interface Template
- Prefix the interface name with “I”

Make the interface public

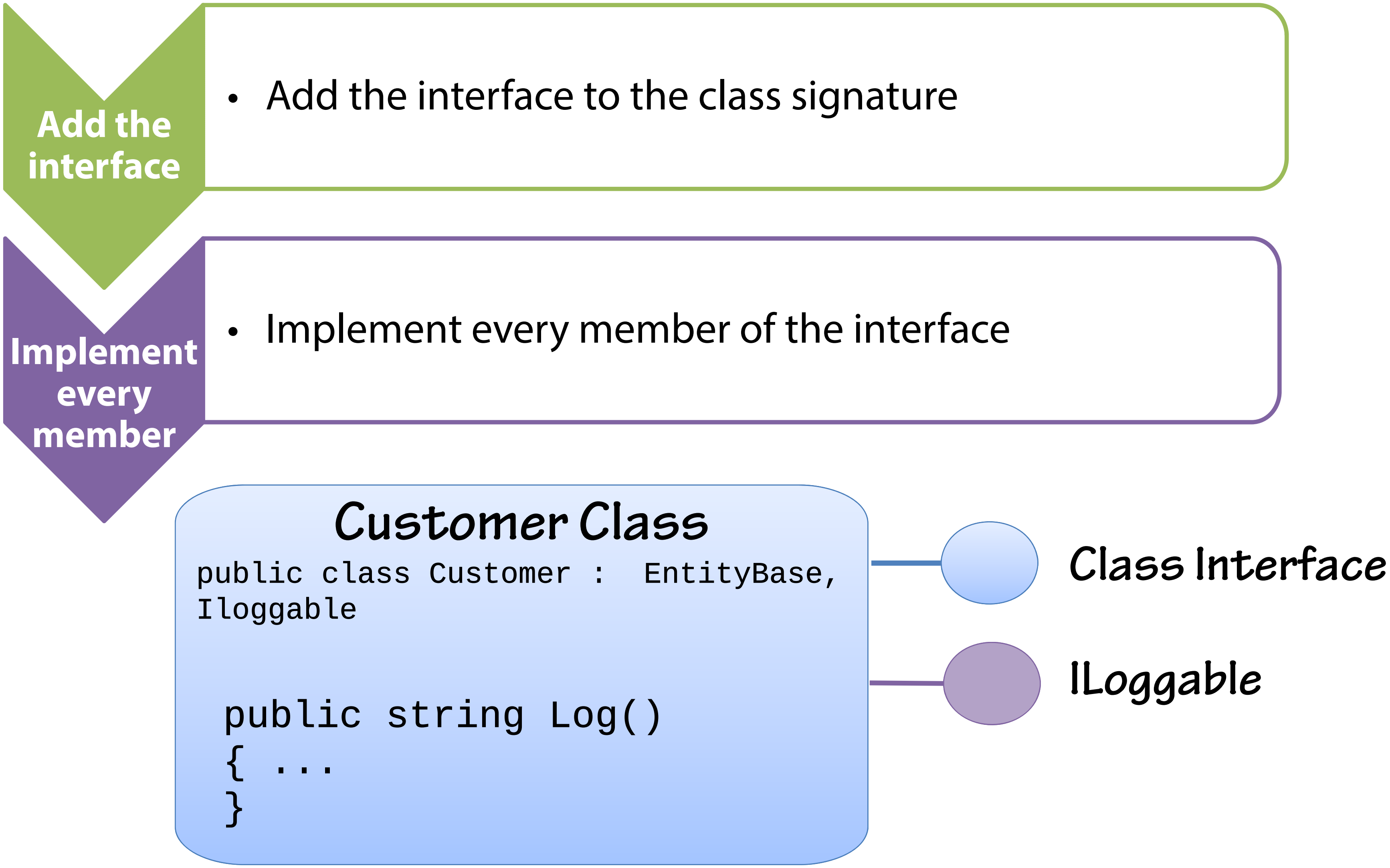
- Add the `public` access modifier

Define members

- Add properties, methods, events, or indexers
- No need for access modifiers on the members
- No implementation of the members.

```
public interface ILoggable
string Log()
```

Implementing an Interface



Add the interface

- Add the interface to the class signature

Implement every member

- Implement every member of the interface

Customer Class

```
public class Customer :  EntityBase,
Iloggable


 public string Log()
 { ...
 }
```

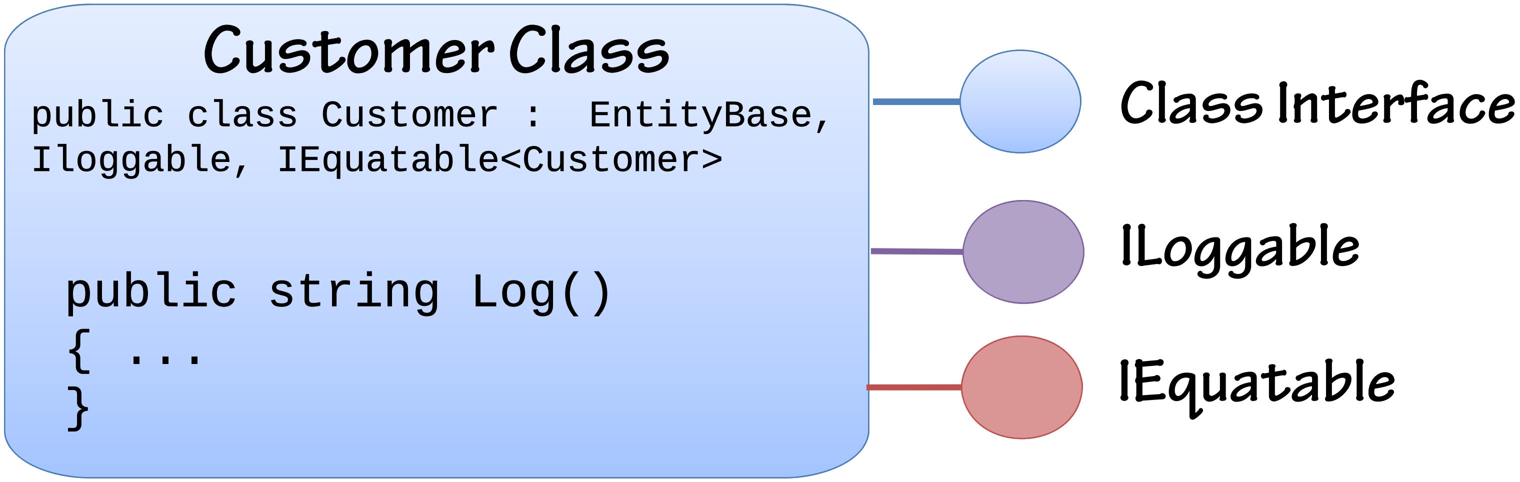
Implementing an Interface

Add the interface

- Add the interface to the class signature

Implement every member

- Implement every member of the interface



Customer Class

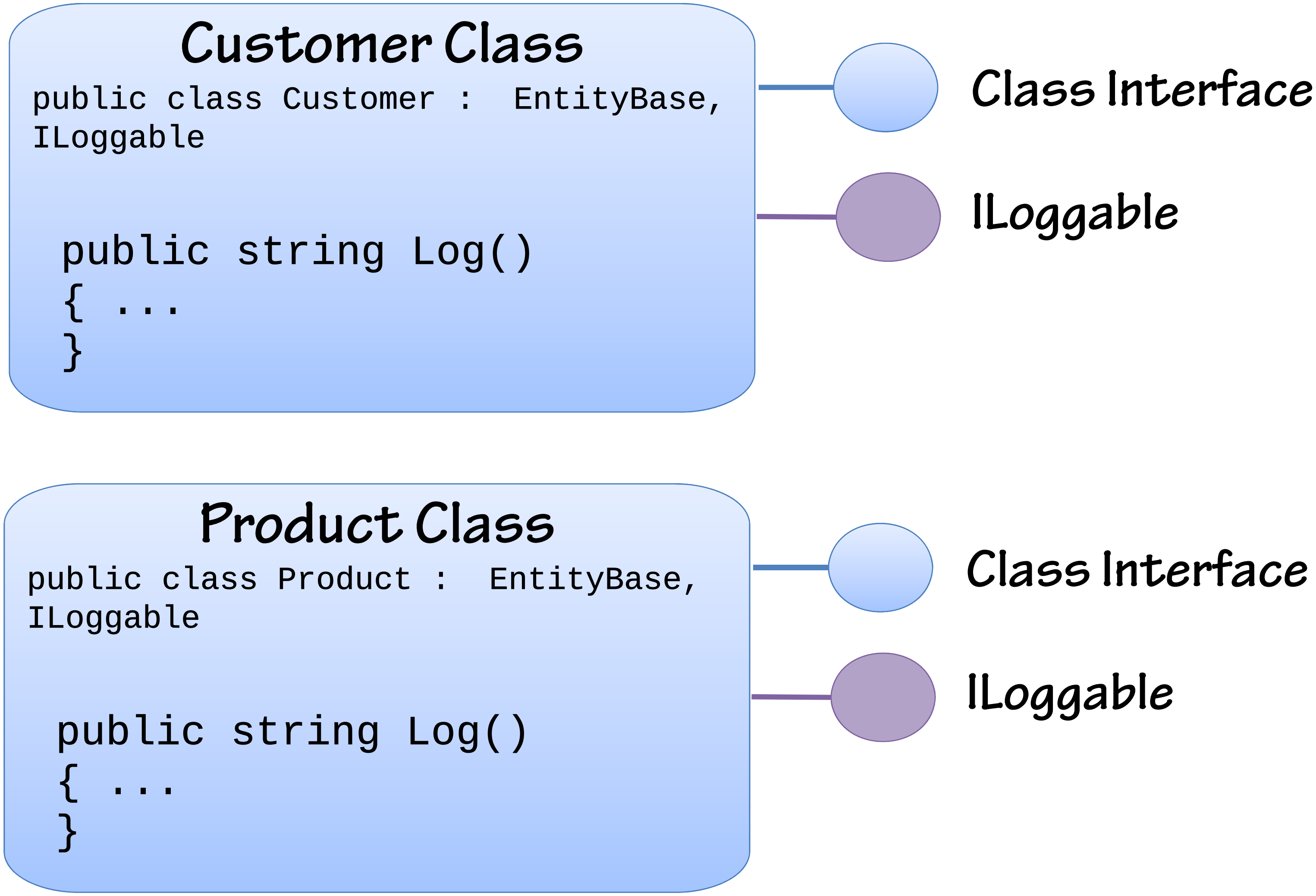
```
public class Customer :  EntityBase,
Iloggable, IEquatable<Customer>

 public string Log()
 { ...
 }
```

Class Interface

ILoggable

IEquatable

Interface-Based Polymorphism



Customer Class

```
public class Customer :  EntityBase,
ILoggable


 public string Log()
 { ...
 }
```

Class Interface

ILoggable

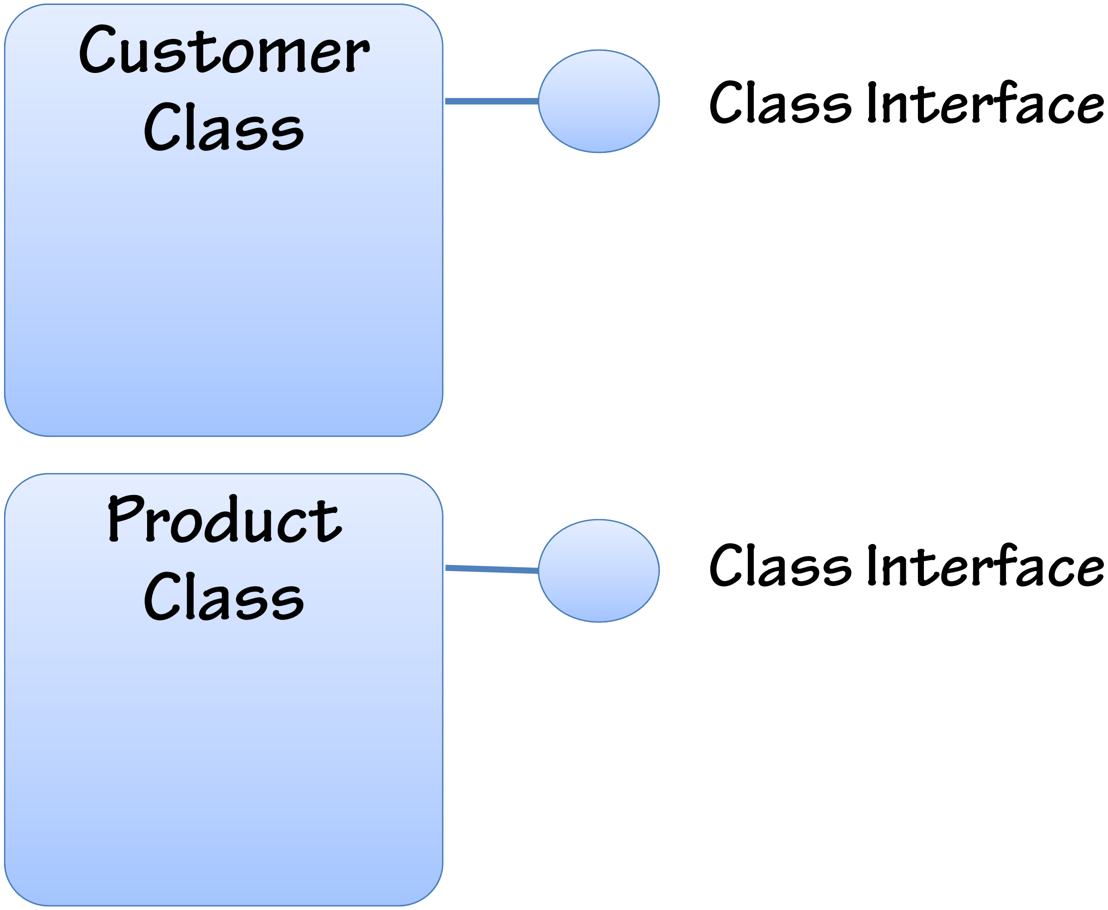
Product Class

```
public class Product :  EntityBase,
ILoggable


 public string Log()
 { ...
 }
```

Class Interface

ILoggable

Summary



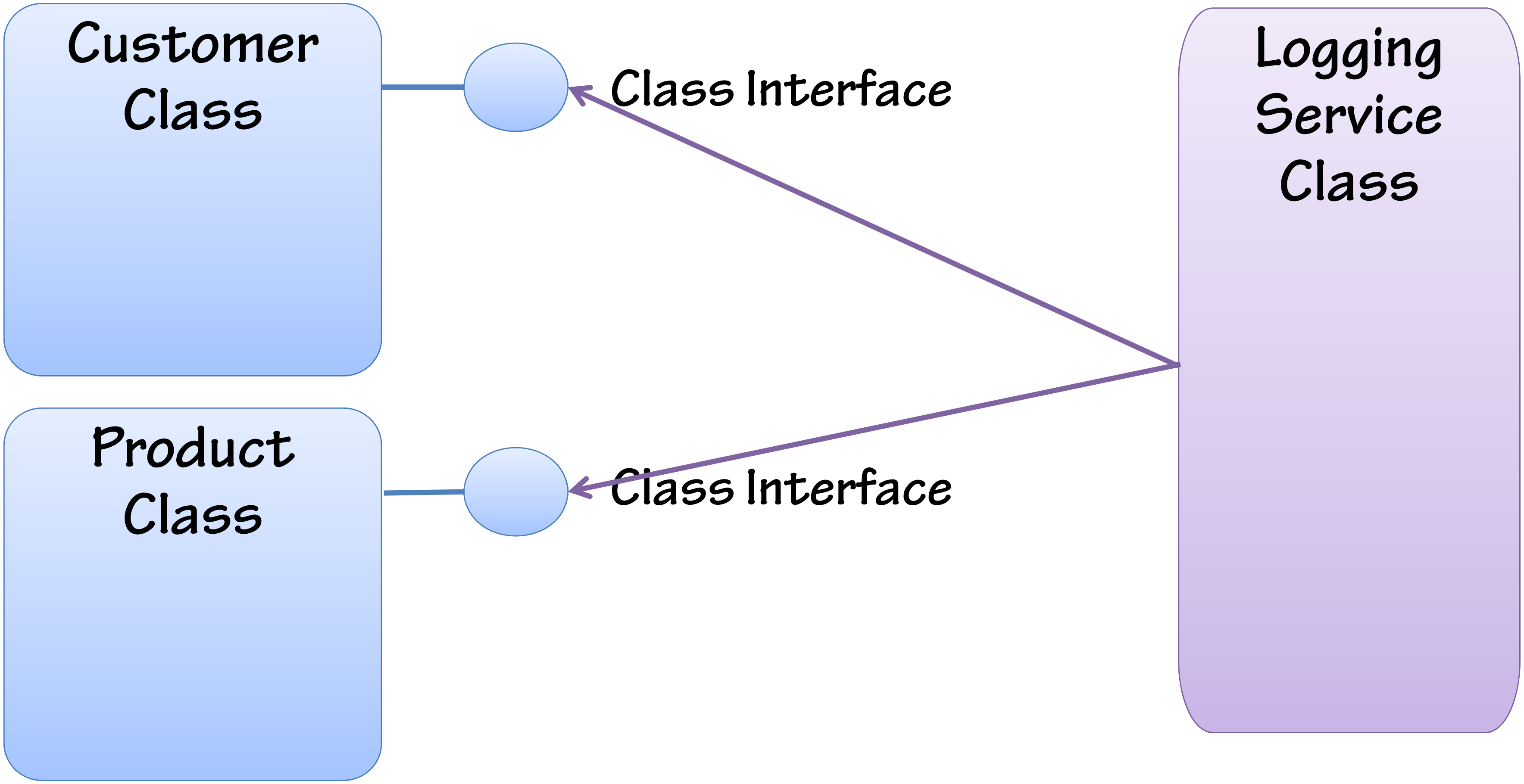
Customer Class

Class Interface

Product Class

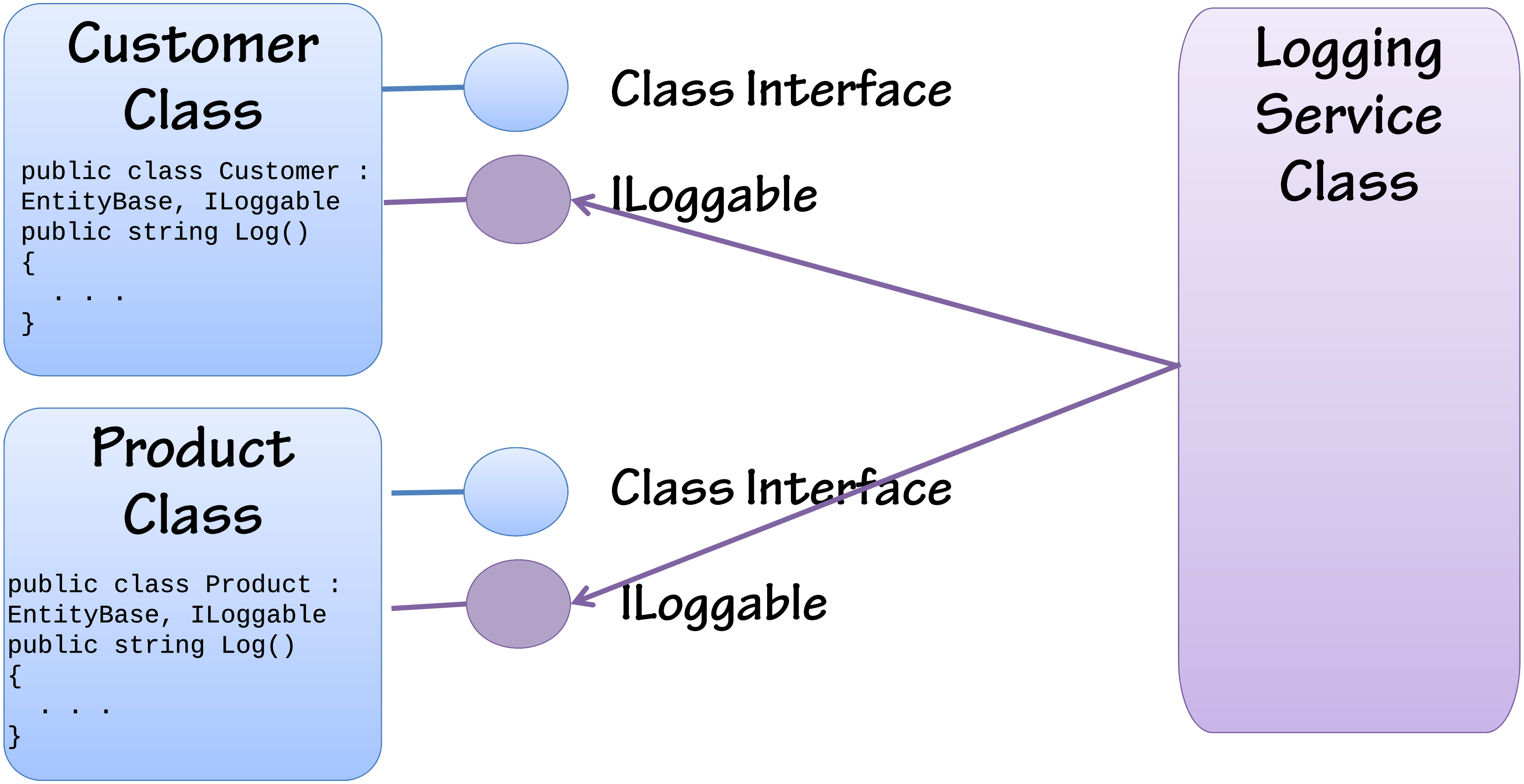
Class Interface

Summary


Customer Class
Product Class
Class Interface
Class Interface
Logging Service Class

Summary

```
public interface ILoggable
string Log()
```



```
public class Customer :
EntityBase, ILoggable
public string Log()
{
  . . .
}
```

```
public class Product :
EntityBase, ILoggable
public string Log()
{
  . . .
}
```