

Generics, Collections, Iterators, and Regular Expressions

Rasmus Lystrøm
Associate Professor
ITU
rnie@itu.dk

```
1 using System;
2
3 namespace BDSA2018.Lecture01
4 {
5     1 reference
6     public class Program
7     {
8         1 reference
9         public static void Main(string[] args)
10        {
11            Console.WriteLine("Hello World!");
12        }
13    }
14 }
```

```
1 using System;
2 using System.IO;
3 using Xunit;
4
5 namespace BDSA2018.Lecture01.Tests
6 {
7     0 references | Run All Tests | Debug All Tests
8     public class ProgramTests
9     {
10        0 references | Run Test | Debug Test
11        [Fact]
12        public void Main_prints_Hello
13        {
14            // Arrange
15            var writer = new StringWriter();
16            Console.SetOut(writer);
17
18            // Act
19            Program.Main(new string[0]);
20
21            // Assert
22            var output = writer.GetStringBuilder().ToString();
23            Assert.Equal("Hello World!", output);
24        }
25    }
26 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\2.1.3\System.Private.CoreLib.dll'. Skipped loading symbols. The debugger option 'Just My Code' is enabled.

Loaded 'C:\Users\rasmus\ Desktop\BDSA2018.Lecture01\BDSA2018.Lecture01\bin\Debug\netcoreapp2.1\BDSA2018.Lecture01.dll'. Skipped loading symbols. The debugger option 'Just My Code' is enabled.

Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\2.1.3\System.Runtime.dll'. Skipped loading symbols. The debugger option 'Just My Code' is enabled.

Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\2.1.3\System.Console.dll'. Skipped loading symbols. The debugger option 'Just My Code' is enabled.

Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\2.1.3\System.Threading.dll'. Skipped loading symbols. The debugger option 'Just My Code' is enabled.

Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\2.1.3\System.Runtime.Extensions.dll'. Skipped loading symbols. The debugger option 'Just My Code' is enabled.

Hello World!

The program '[15304] BDSA2018.Lecture01.dll' has exited with code 0 (0x0).

OUTLINE

.NET Core Launch (console) (BDSA2018.Lecture01) BDSA2018.Lecture01.sln

Ln 21, Col 1

Agenda

Generics

Iterators, enumerators, and enumerations

Collections

Regular Expressions

Generics

“Parametric Polymorphism”

Built-in

Create your own?

Type Constraints

(Co- and contravariance)

ArrayList → List<T>

// Non-generic

```
IList list = new ArrayList();
```

```
list.Add("hello");
```

```
var s = (string)list[0];
```

// Generic

```
IList<string> list = new List<string>();
```

```
list.Add("hello");
```

```
var s = list[0];
```

Create your own generic class

```
public class MyStack<T>
{
    public void Clear() { }

    public T Peek() { }

    public T Pop() { }

    public void Push(T item) { }
}
```

Create your own generic method

```
public string Serialize<T>(T obj) {}
```

Type constraints

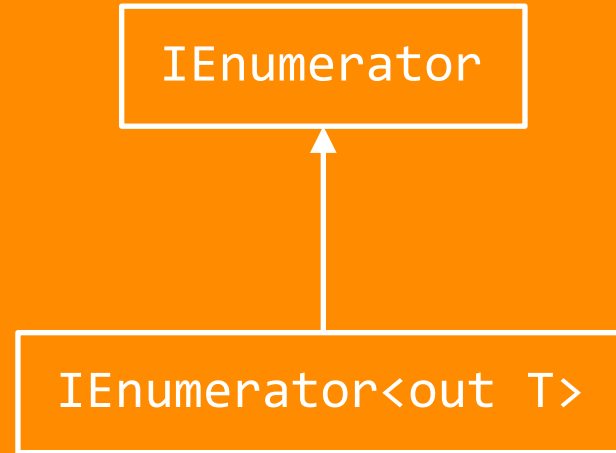
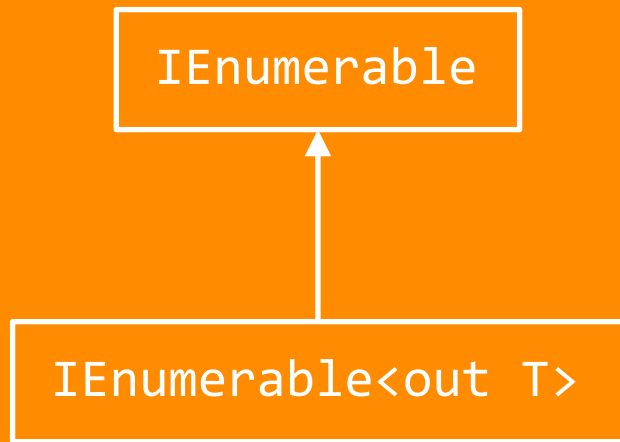
```
public class MyConstrainedGenericClass<T>  
    where T : class { }
```

```
public class MyConstrainedGenericClass<T>  
    where T : struct { }
```

```
public class MyConstrainedGenericClass<T1, T2>  
    where T1 : Foo where T2 : IBar { }
```

```
public T2 MyConstrainedMethod<T1, T2>(T1 item)  
    where T1 : Foo  
    where T2 : IBar { }
```

Iterators



Producer

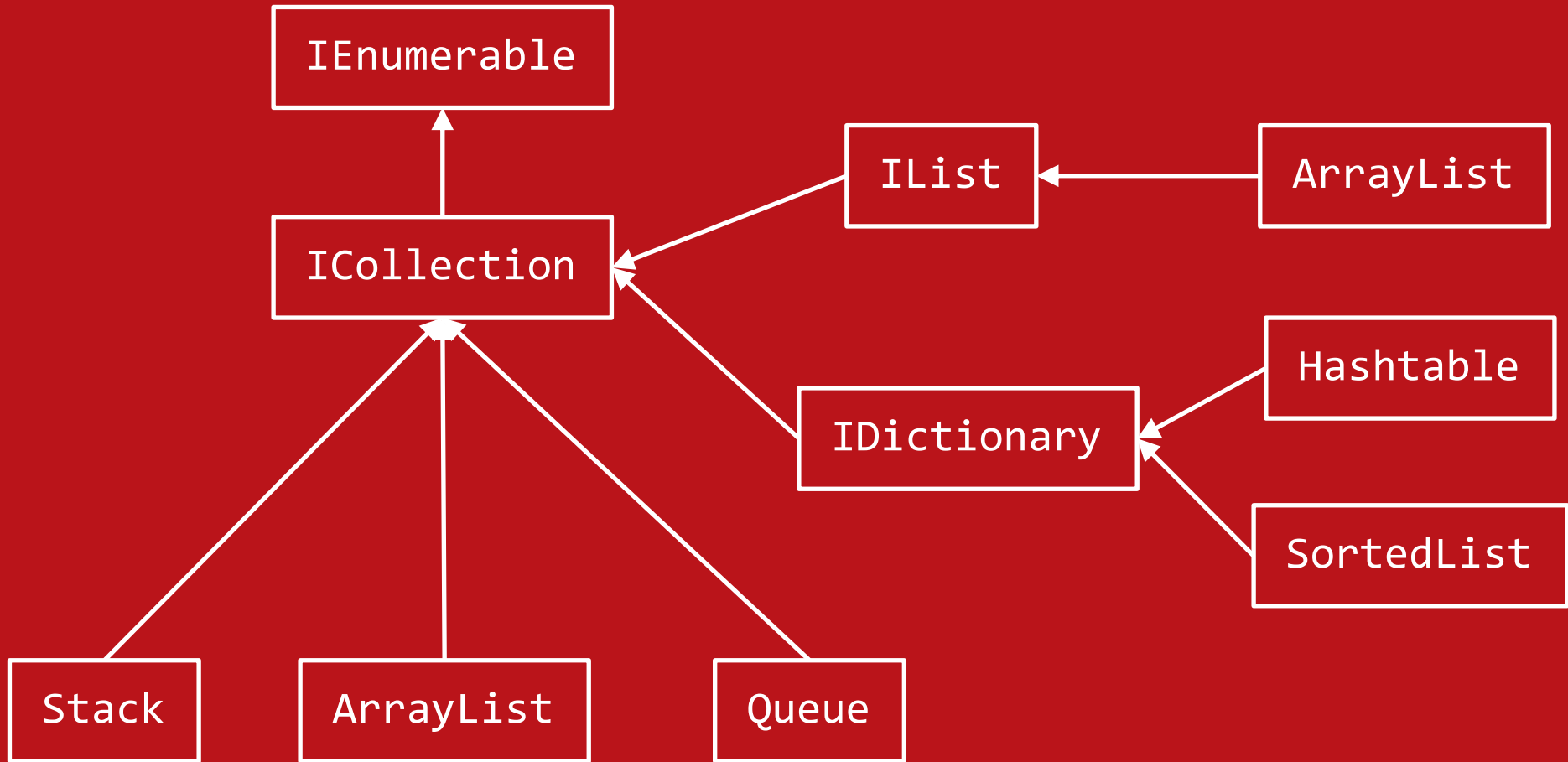
`yield return T;`

`yield break;`

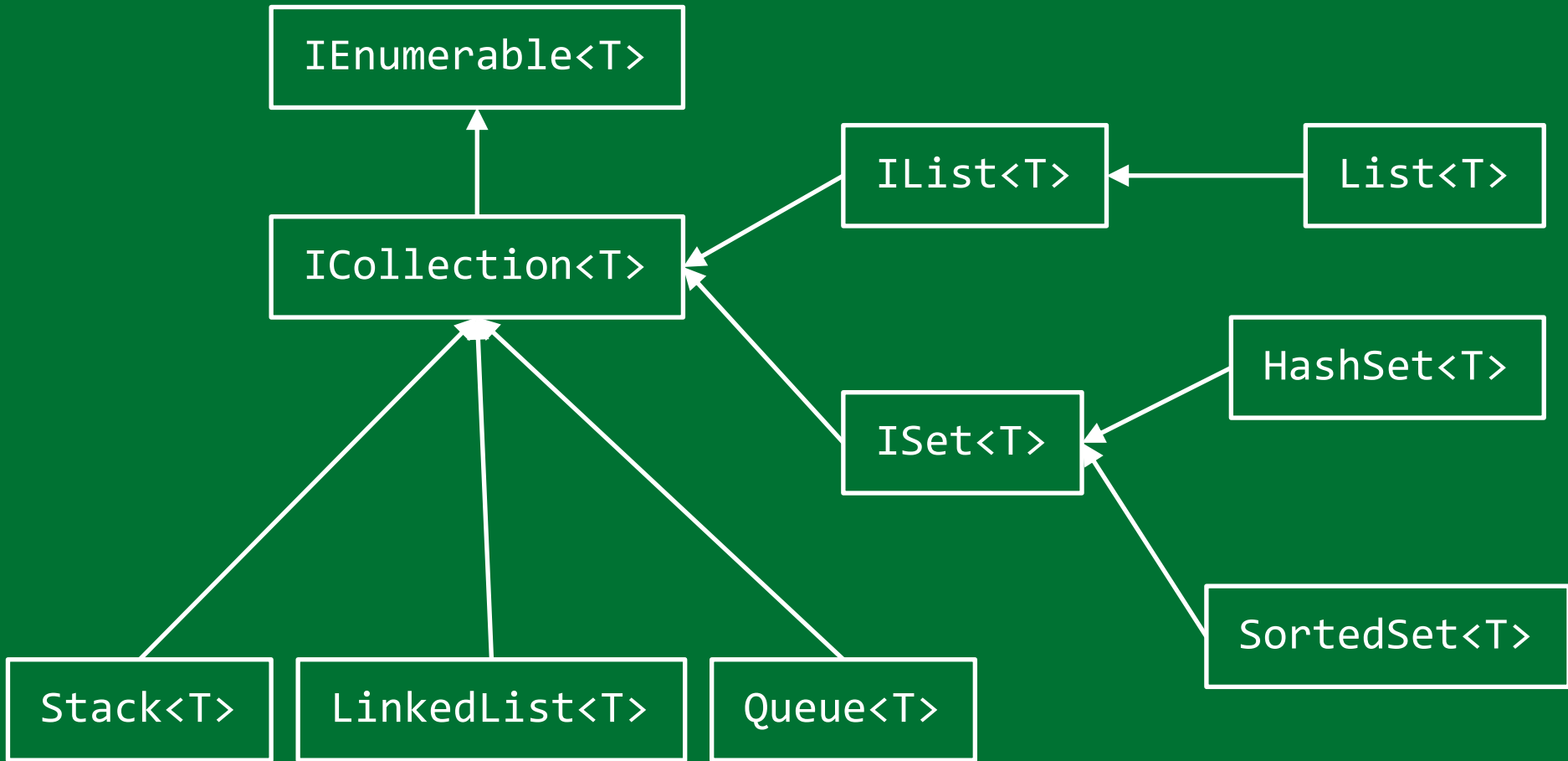
Building block for Linq

```
foreach (var item in items)
{
}
```

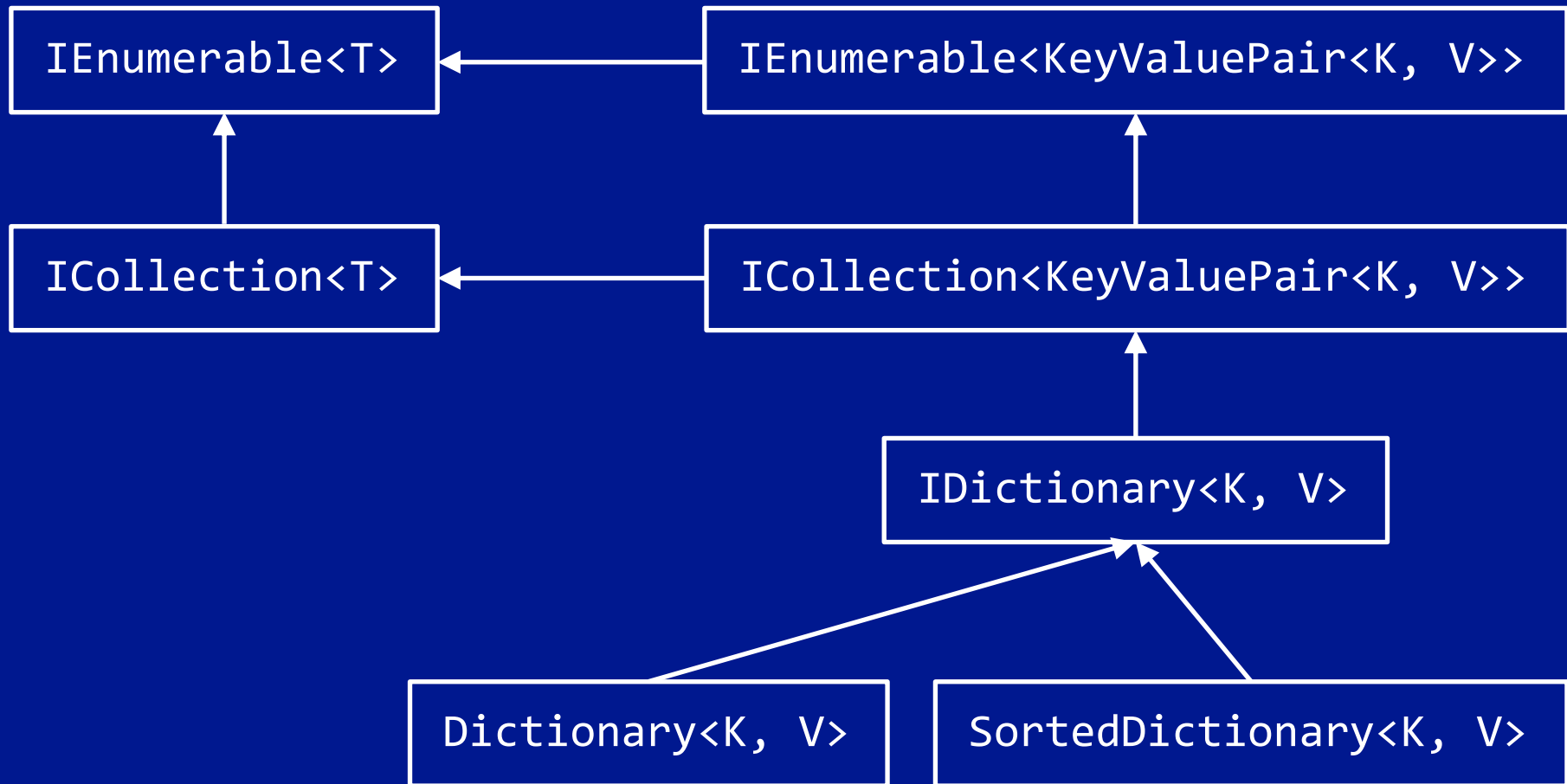

System.Collections



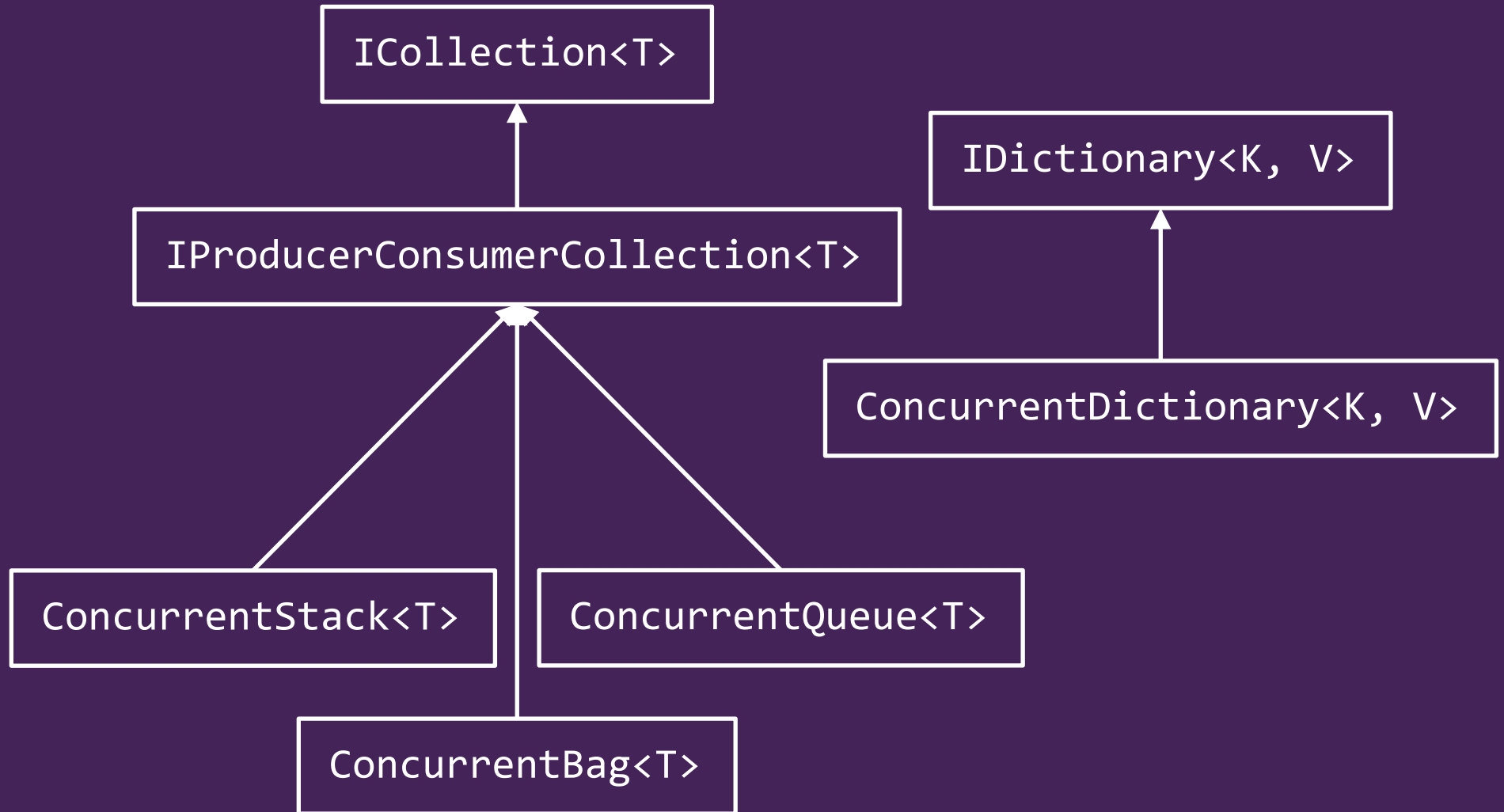
System.Collections.Generic



System.Collections.Generic 2



System.Collections.Concurrent



Regular Expressions

*	Zero or more times the previous character
+	Once or more times the previous character
?	Zero or one time the previous character
.	Any single character (not \n)
\s	Any whitespace character (e.g. tab)
\S	Any non-whitespace character
\b	Word boundary
\B	Any non-word boundary position
\w	Any word character (a-z, A-Z, 0-9)
\W	Any non-word character
^	Start of the input text
\$	End of the input text

Regular Expressions

[1c]	matches character '1' or 'c'
[a-z]	matches all lower-case letters
[a-zA-Z]	matches all letters
[0-9]+	matches integer numbers
[0-9]+\.[0-9]+	matches a floating point
[0-2][0-9]:[0-5][0-9]	matches a time e.g. 12:34