Understanding Value Types and Reference Types

Gill Cleeren
CTO Xpirit Belgium

@gillcleeren | www.xpirit.com/gill

Agenda

The Common Type System

Understanding custom types

Creating enumerations

Working with struct

The Common Type System

The Common Type System

.NET specific

Standard

Defines how type definitions and values are handled in memory

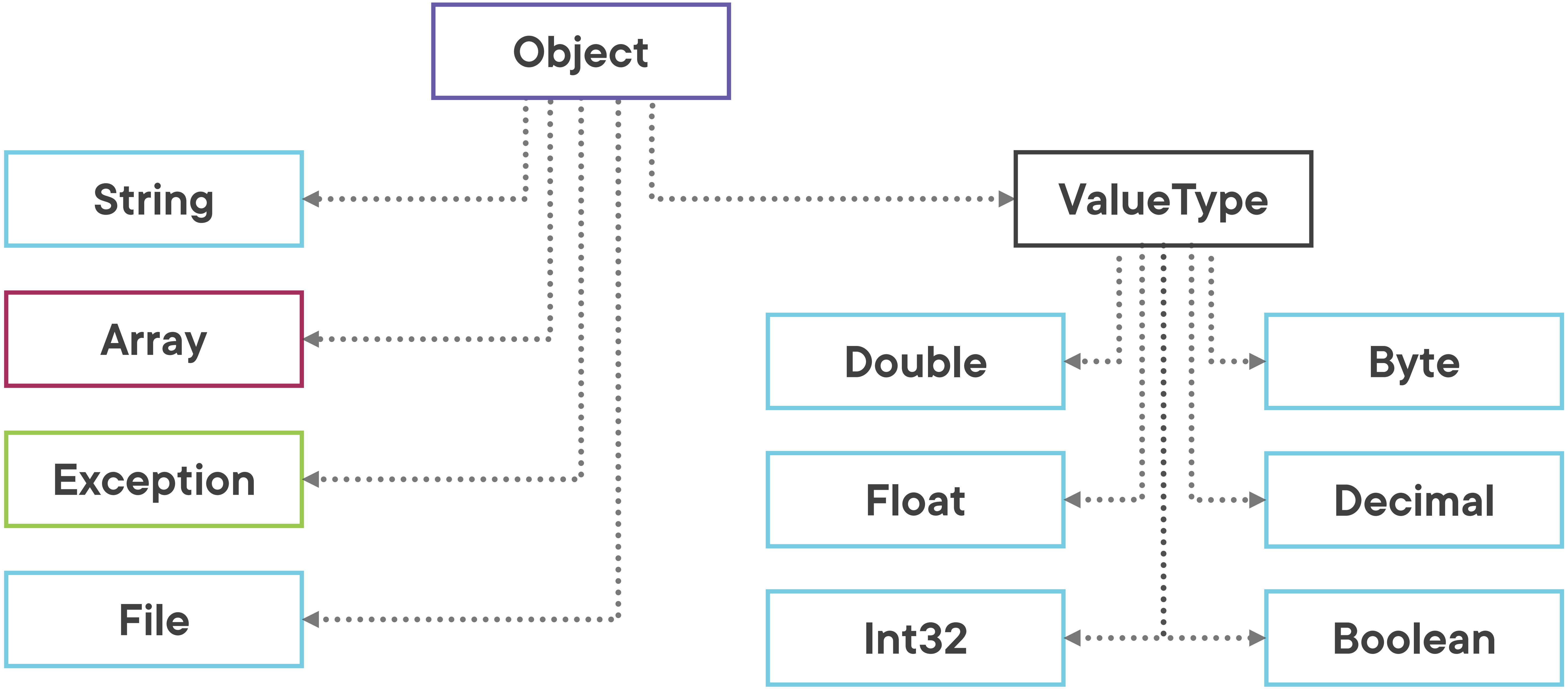
Shared across multiple languages including C#

Types in .NET and C#

Value types

Reference types

The Type Hierarchy



Everything is an object

Types in the CTS

Enumeration

Struct

Class

Interface

Delegate

Types in .NET and C#

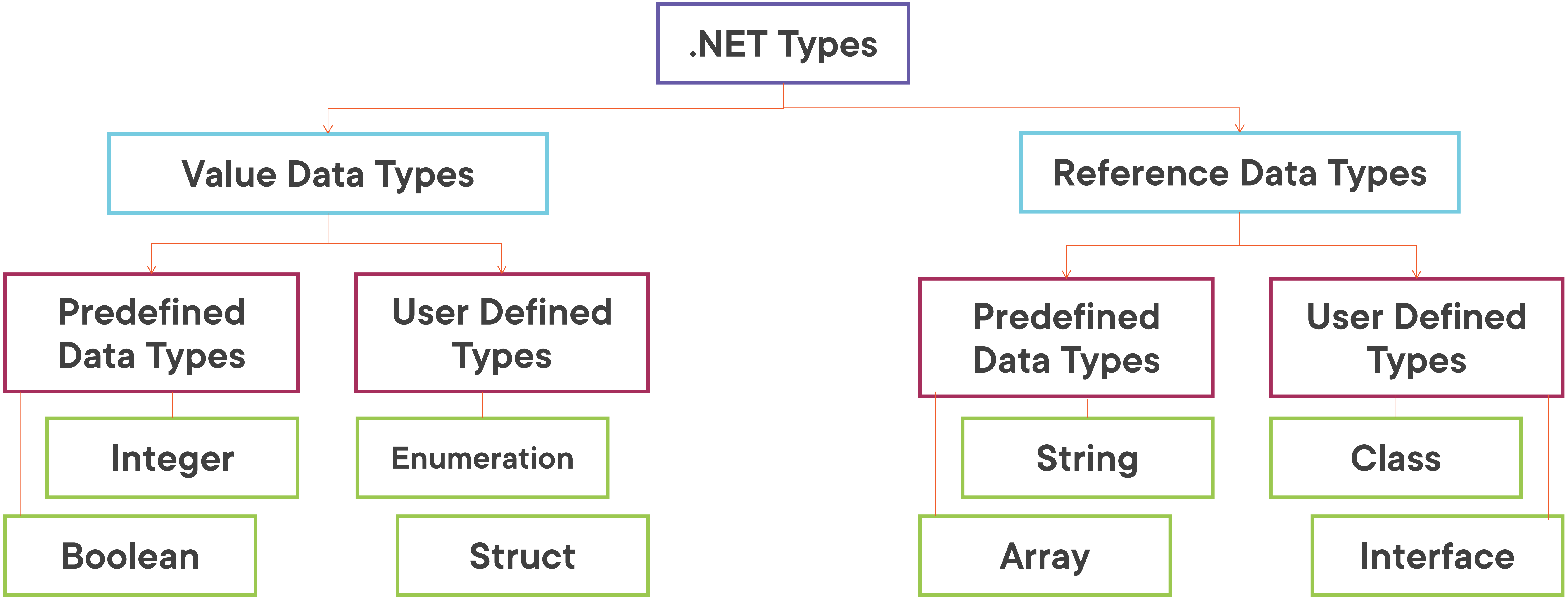
Value types

Enumerations & Structs

Reference types

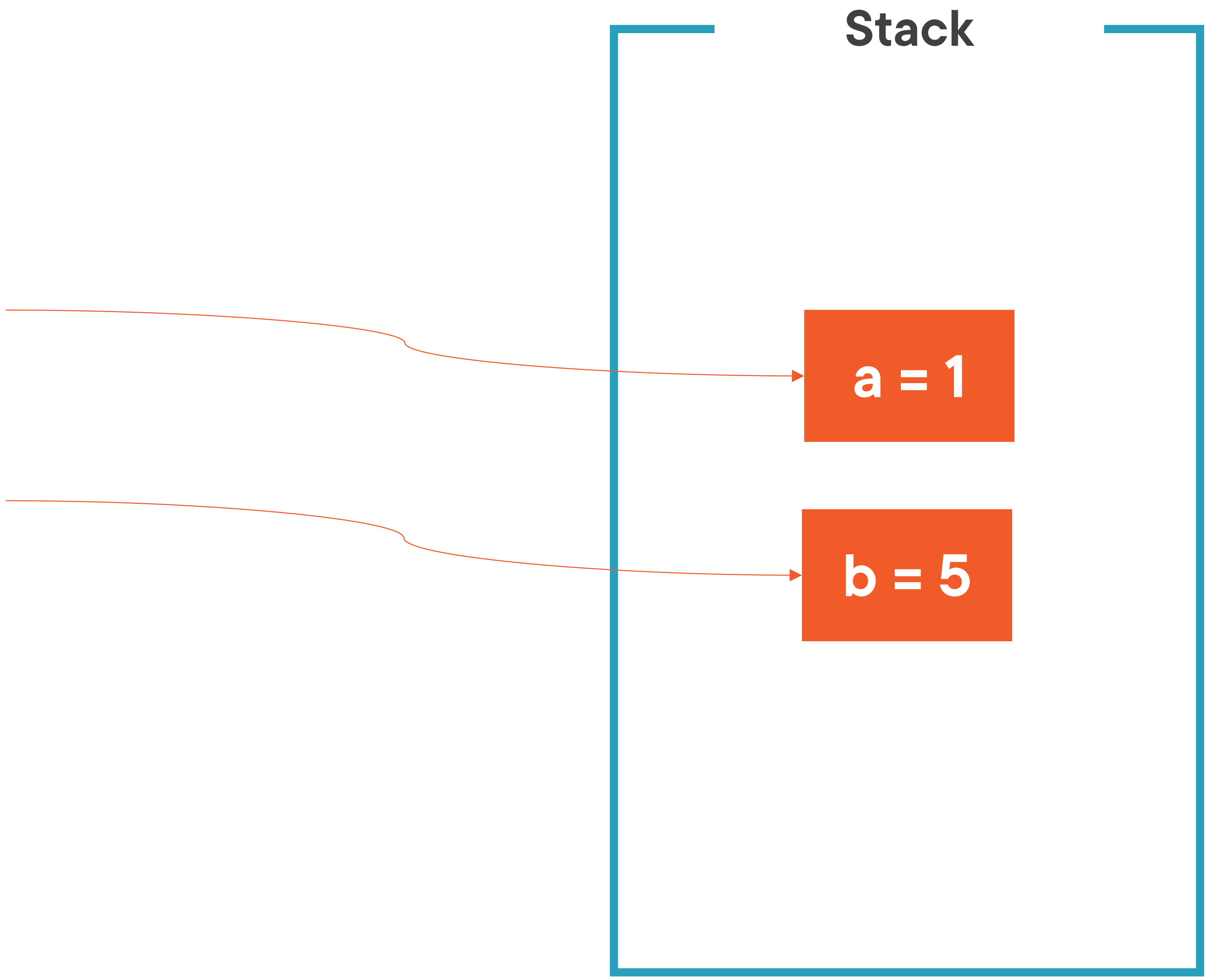
Classes, Interfaces & Delegates

Types in .NET and C#

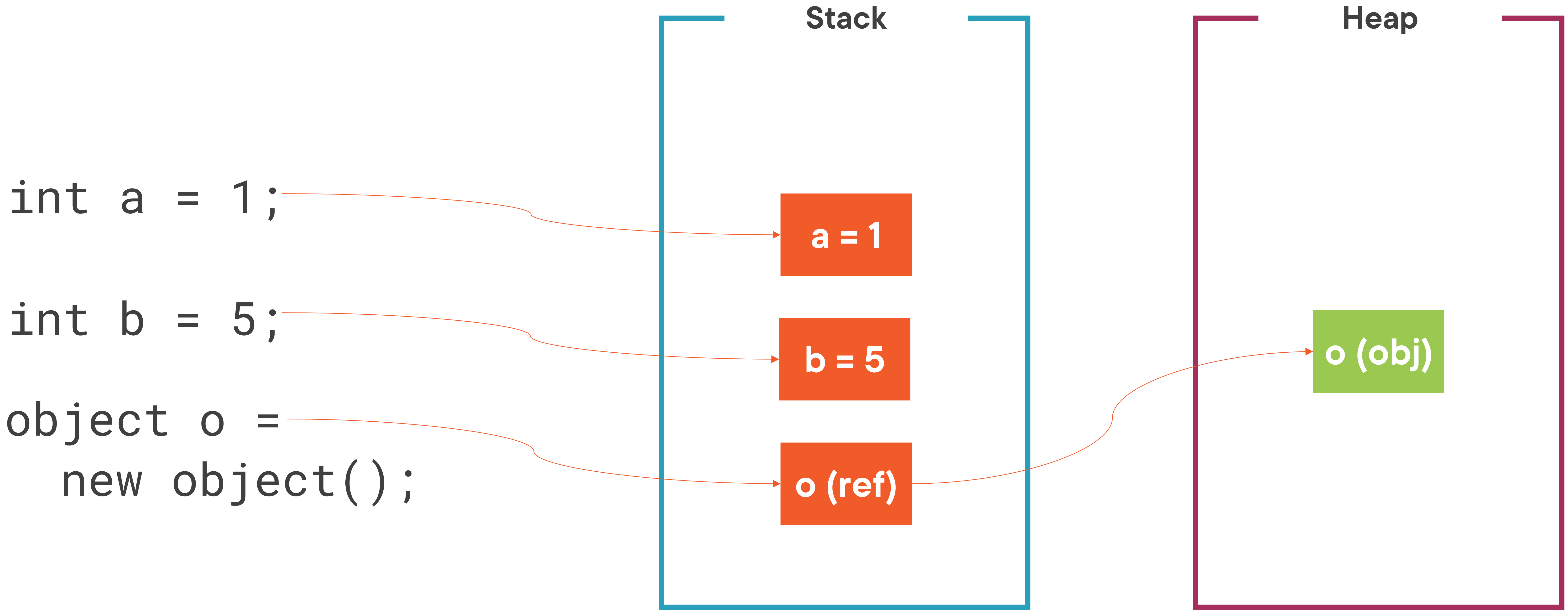


Working with Value Types

```
int a = 1;

int b = 5;
```



Working with Reference Types



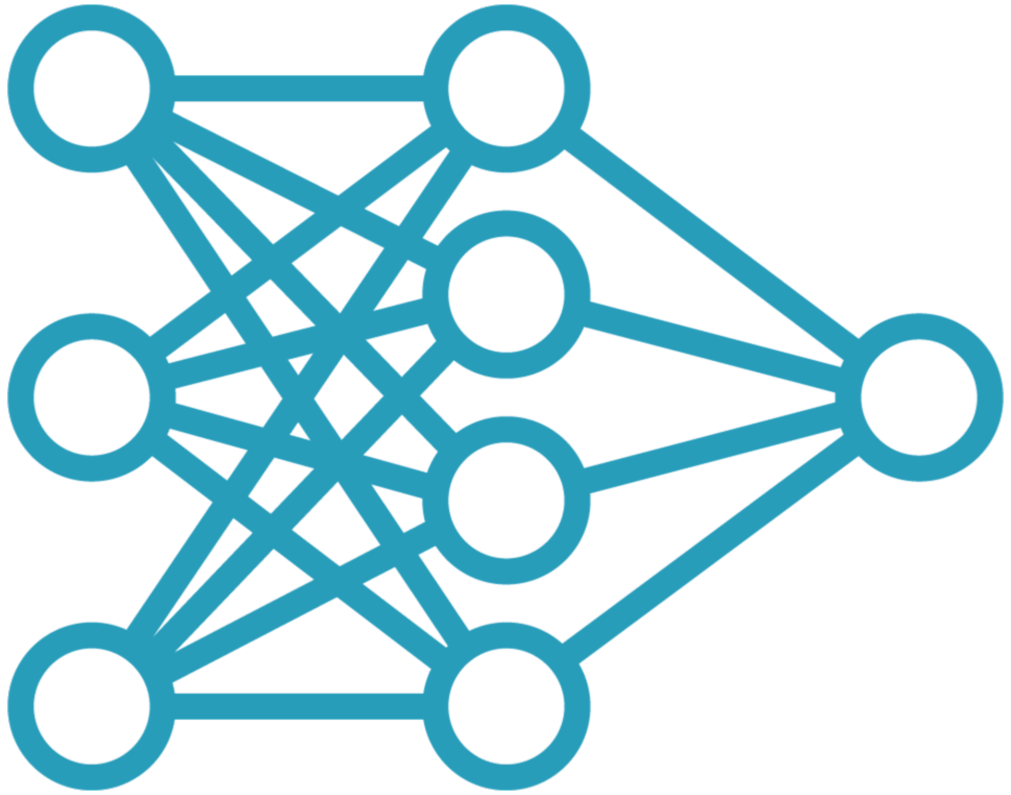
```
int a = 1;

int b = 5;

object o =
   new object();
```

Understanding Custom Types

With just variables, we only get so far.

If we want to represent a structure, we need a custom type.

Custom Types



Class

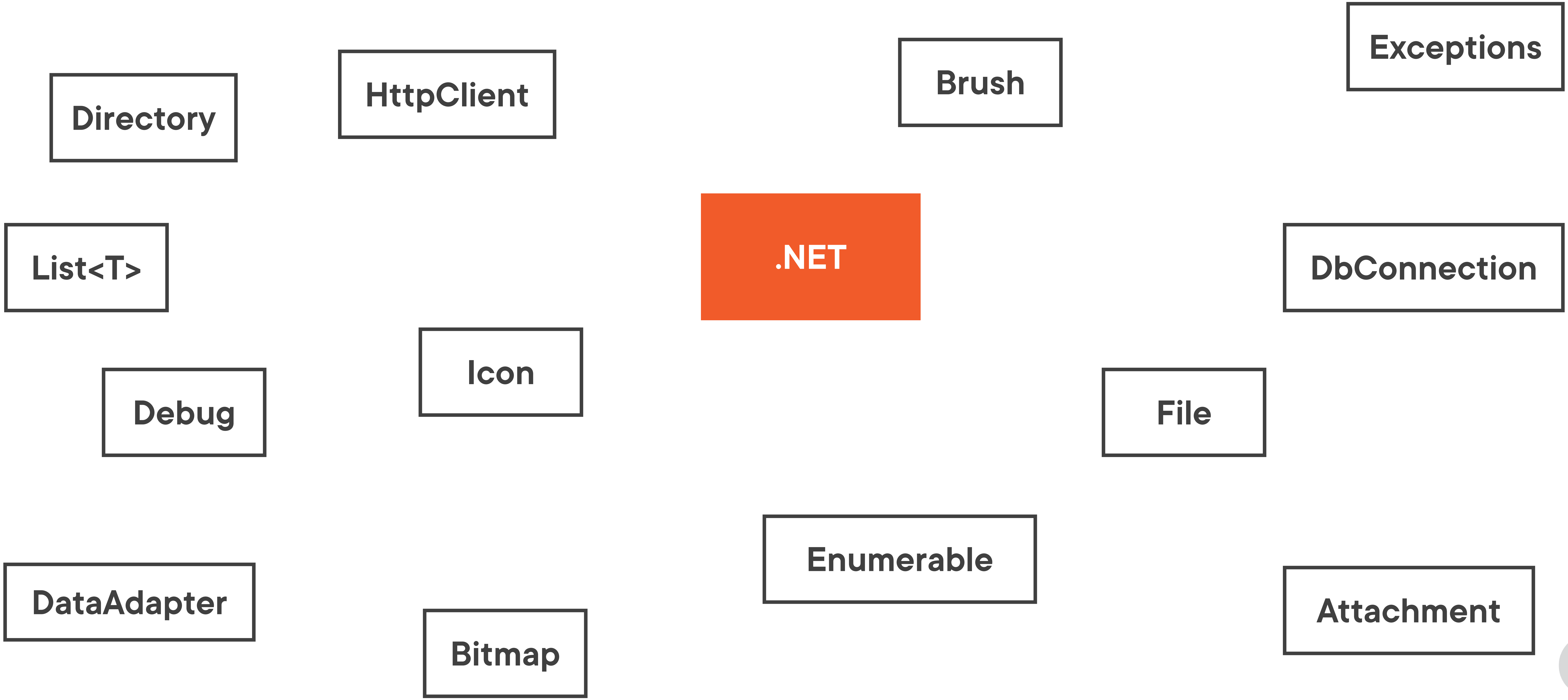
Reference type

Most commonly used

Struct

Value type

Custom Types in .NET



Working with Custom Types

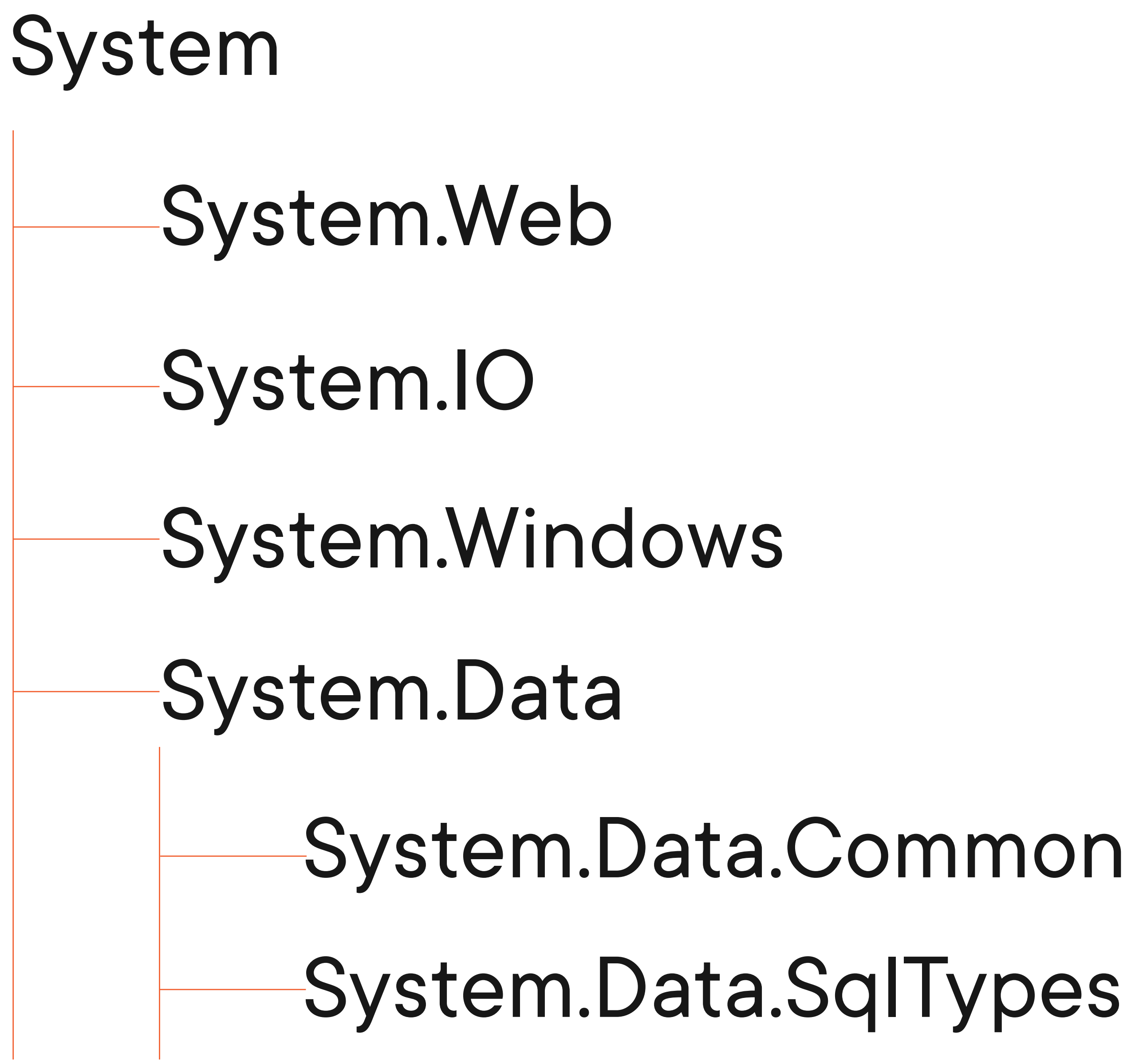
Contain data

Define their functionality

.NET library contains many types

Some are known, some need to be imported through assembly ref

Organizing Types in Namespaces



System

System.Web

System.IO

System.Windows

System.Data

System.Data.Common

System.Data.SqlTypes

```
using System;

using System.Data;


DataTable dataTable = new DataTable();//available through System.Data namespace
```

The using Keyword

A using Statement only brings the types within the specified namespace, not the ones in nested namespaces

Demo

Browsing for existing types

Using a custom type

Creating Enumerations

Using an Enumeration in C#

Named constants for improved readability

Value type

Uses enum keyword

```
enum EmployeeType
{

        Sales,

        Manager,

        Research,

        StoreManager

}
```

Creating an Enumeration

```
enum EmployeeType
{

        Sales,//0

        Manager,//1

        Research,//2

        StoreManager//3

}
```

Default Values for Enumerations

```
Console.WriteLine(EmployeeType.Sales);
```

Accessing Enum Values

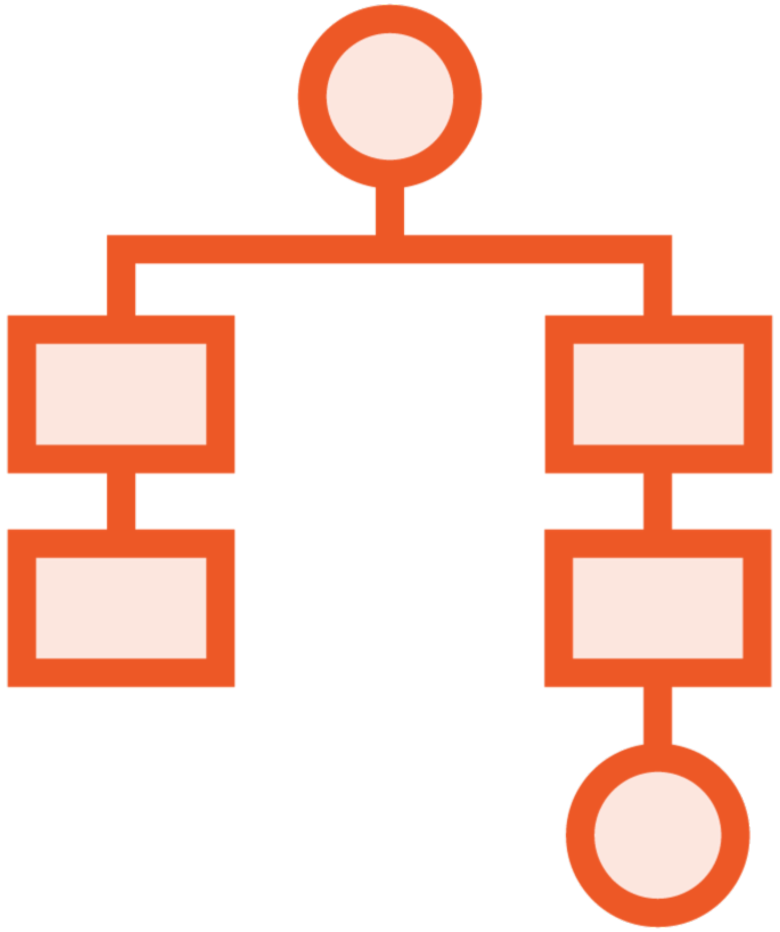
Demo

Creating an enumeration

Using the enumeration

Accessing the values

Working with Struct



Creating a Struct

- Represents a custom data structure
- Value type
- Can be new'ed
- Can contain methods and other members

Declaring a Struct

```
struct Employee
{

    public string Name;

    public string Department;

}
```

Using a Struct

Struct name

Variable name

```
Employee employee;

employee.Name = "Bethany";

employee.Department = "Sales";
```

Adding a Method to the Struct

```
struct Employee
{

    public string Name;

    public string Department;

    public void GetPaid()
    {

        //Code to pay out wage

    }

}
```

Demo

Creating a struct

Using the struct

Summary

Types are the most essential building block in C#

CTS defines several custom options

Enumerations and structs offer ability to create custom types

Resources

Other relevant courses in the C# path:

- Object Oriented development in C#
  - Deborah Kurata

Up next:

Creating classes and objects