Error Handling

Chapter 10 – Professional C#

https://csharp.christiannagel.com

Topics

- Exceptions…
- try/catch
- try/finally
- Exception Filters
- Caller Information

Exception Class

- Error Number
- Message
- Source
- Stack Trace

Predefined Exception Types

- ArgumentException
- ArgumentNullException
- ArgumentOutOfRangeException
- NotSupportedException
- StackOverflowException
- OverflowException
- FileLoadException
- TaskCanceledException
- …

User Defined Exception Types

- Derive from Exception base class
- or any other predefined Exception type

try/catch/finally

```
try
{
  // code for normal execution
}
catch
{
  // error handling
}
finally
{
  // clean up
}
```

Multiple Catch Blocks

```
try
 {
   //...
   string? userInput = Console.ReadLine();
   //...

   int index = Convert.ToInt32(userInput);

   if (index < 0 || index > 5)
   {
     throw new IndexOutOfRangeException($"You typed in {userInput}");
   }
 }
 catch (IndexOutOfRangeException ex)
 {
   //...
 }
 catch (Exception ex)
 {
   //...
 }
```

Exception Filters

- Filter exception to not catch all exceptions of a type

```
try
{
  ThrowWithErrorCode(405);
}
catch (MyCustomException ex) when (ex.ErrorCode == 405)
{
  Console.WriteLine($"Exception caught with filter {ex.Message} " +
    $"and {ex.ErrorCode}");
}
catch (MyCustomException ex)
{
  Console.WriteLine($"Exception caught {ex.Message} and {ex.ErrorCode}");
}
```

Re-throwing Exceptions

- Use throw instead of throw ex
- Changing the exception: assign the inner exception

Caller Information

- CallerLineNumber
- CallerFilePath
- CallerMemberName

Guidelines

- Don’t catch exceptions of type Exception without rethrowing
- Don’t throw Exception, use derived types
- Use *throw* instead of *throw ex*