Kursträff 8: Pair programming exercises - Solutions

Use the course literature and the slides from today's lecture to discuss and answer the questions below.

1. Discuss the following terms and keywords -- what do they mean and how are they used?
   - Exception
     A class from which all other Exception-types inherit
   - throw
     Used to throw an exception: throw new Exception();
   - catch
     Used to catch a thrown exception
   - finally
     An optional part of a try/catch block that is always executed last in the try/catch-chain, whether an exception was thrown or not
   - using-block
     A simplified way of working with classes that implement IDisposable. Equivalent of try/finally
   - IDisposable
     An interface which is implemented by all classes that can be used with using-blocks
2. Give an example of a try/catch-block that uses int.Parse to convert a string to int and which uses Console.WriteLine to write the exception-message to the console if an exception is thrown

```
try {
   var n = int.Parse("123m");
} catch (Exception ex) {
   Console.WriteLine(ex.Message);
}
```

3. C# has a class called StringReader, which can be used to read from strings as if they were streams.
    - Create a new console application
    - In the Main()-method:
        - Create a new StringReader object from a verbatim string containing a few lines of text
            - Use a using block
        - Read the lines of the StringReader one at a time and output them to the console

```
var text = @"This is a pretty
long piece of text.";
using (var sr = new StringReader(text)) {
   var line = sr.ReadLine();
   while (line != null) {
      Console.WriteLine(line);
      line = sr.ReadLine();
   }
}
```

4. Download and unzip the example-project BookSerializer.zip from Blackboard
    - Open the solution in VisualStudio
        - Go through the file structure together and discuss what the various files and classes in the project do
            - Look at the files Program.cs, Interfaces/ISerializeable.cs, and Models/SerializeableList.cs and discuss what they do and how they are used
    - As you may have noticed, the application doesn’t function very well. There are several sections marked “TODO” in the code, in:
        - Program.cs
            - Write exceptions are unhandled
        - Data/SerializedReaderWriter.cs
            - Both the Read and Write methods are missing the code to load and save data
        - Models/Book.cs
            - The implementation of Deserialize() needs to be completed
        - Models/SerializeableList.cs
            - Exceptions aren’t handled in the Deserialize()-method
    - Discuss and fix the problems above

A separate solution will be uploaded to Blackboard