1. **Player / IPlayer**
   1. “Player” doesn’t match C# conventions for interface naming. Will title IPlayer instead.
   2. Interfaces can’t define a constructor. Inherited classes can though! As such, I’ll add an abstract class (called “Player”) that implements IPlayer, and enforces a constructor that accepts a string.
      1. HumanPlayer and ComputerPlayer will, therefore; inherit from the abstract Player class, rather than the IPlayer interface directly.
         1. Any ‘player’ logic that needs doing will be done through the IPlayer interface, thereby allowing direct usage of the IPlayer interface in the future.
      2. Due to both Player and IPlayer both being so small and tightly coupled, they will be in the same file (Player.cs)
   3. Added make move method to interface. Method stub below:
      1. bool MakeMove(Nim \_game);
2. **Difficulty**
   1. For ease-of-use purposes, I’ve added a static class of extension methods for the Difficulty enum. This allows us to pretend that these enums are objects when the distinction doesn’t matter, and centralizes and standardizes the location where difficulty settings are mapped to game logic changes.
      1. Due to such tight coupling, this static class is in the same file as the enum (Difficulty.cs)
3. **Nim**
   1. Added private “CommonCtor” method to reduce redundant code in constructors.
   2. Converted Winner to a property, rather than a method. Otherwise identical.
   3. Converted IsGameOver to property rather than method, and renamed it to “GameOver” (no proceeding “Is”). Otherwise identical.
   4. Removed ComputerMakeMove. That behavior should belong to the player making the move.
4. **NimSetupView**
   1. Added failure feedback
5. **HeapUserControl**
   1. Description for constructor makes no sense, so I’m ignoring it.
6. **Heap**
   1. Added backlink to HeapUserControl, making the coupling two-way.
7. **NimGamePage**
   1. Removed RemoveStones method, as it’s redundant to Nim’s RemoveStones method.
   2. Moved setup with the backend Nim game to a separate function, rather than in constructor for ease of assembly.
   3. Changed stub of ValidateInput. Before and after below:
      1. **Before:** bool ValidateInput();
      2. **After:** bool ValidateInput(out int heap, out int quantity);