# Adding Exception Handling to the SmartARray

**The goal for this exercise** is to make sure that you can use an exception handling. You will also modify the SmartArray class – so much so that you'll actually put all this code into a new project.

**What you need to do for this exercise:**  In the starter solution, in the Program\_SmartArray\_EH\_Test class, add (and/or modify) any code that you need to in order to re-implement the SmartArray class so that it uses the new UnderflowException and OverflowException classes, as defined below. You are encouraged to copy code from your existing SmartArray class (from previous lessons, for example)

As you create the new SmartArray class (and later, the Stack, Queue and LinkedList classes), make sure that you use exception handling as your return values for the methods you create (as appropriate).

In the table below, you'll see that stuff that hasn't changed has been grayed out, stuff that has changed has been marked in red.

You will also need to modify the test code that was previously provided, in order to test out your changes. You can do this using the test code that is included in the provided starter project.

|  |  |  |
| --- | --- | --- |
| **SmartArray Data & Methods** | | |
| Data Field Name | Type | Description: |
| rgNums | Array of ints | A reference to an array of integers. |
| Note: all data fields should be marked **private** | | |
| Method Name | Returns | Description/Parameters: |
| Default constructor | <nothing> | Allocates an array of 5 integers.  The rest of the class must be written so that if we were to change this to be, say, 10 integer, all other methods would function still correctly. |
| **Second, overloaded constructor** | **<nothing>** | **Parameters:**   1. **An integer that specifies what the size of the array should be**   **Allocates an array (the size is given by the parameter).** |
| SetAtIndex | <nothing> | Parameters:   1. An integer that is the index of the element to set 2. An integer that is the value to set that element to   Conceptually,  **SmartArrayObject.SetAtIndex(10, 20)**  this will accomplish the same sort of thing as saying  **Array[10] = 20;**  would accomplish with a normal array  This will throw an UnderflowException if the index is too low, and will throw an OverflowException if the index is too high. These exceptions are defined in the Program.cs file for this project. |
| GetAtIndex | The integer value at the array slot. | Parameters:   1. An integer that is the index of the element to get   Conceptually,  **SmartArrayObject.GetAtIndex(10);**  this will accomplish the same sort of thing as saying  **Array[10];**  would accomplish with a normal array  This will throw an UnderflowException if the index is too low, and will throw an OverflowException if the index is too high. These exceptions are defined in the Program.cs file for this project. |
| PrintAllElements | Nothing (void) | Parameters: None  Print all elements of the array, one per line, onto the console. |
| Find | **true** if at least one element in the array is the same value as the parameter, **false** if the given value isn't present in the array. | Parameter:   1. An integer that may or may not be in the array.   This method takes it's parameter, and sees if that value is located anywhere within the array. If it finds even a single slot of the array with the same value as the parameter, it will return true. If the value isn't found anywhere within the array, it will return false. |
| Note: all methods should be marked **public** | | |