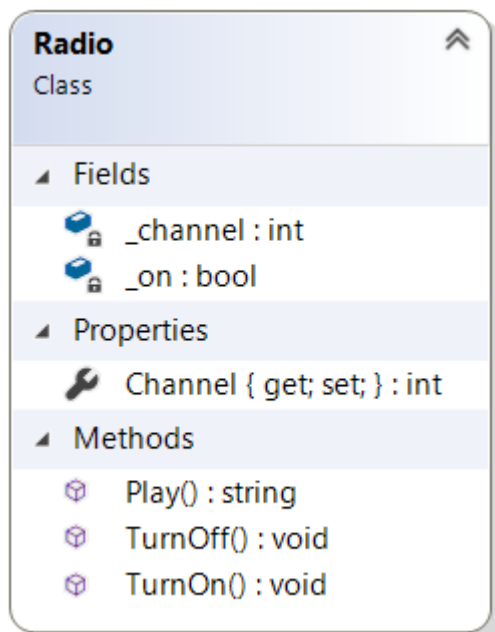# Radio Mini-Project

Set-up

Download the starter code Radio_Mini_Project.

The Radio_Mini_Project solution contains two projects:

- RadioApp, which contains an empty Radio class
- RadioTests, which contains unit tests for the Radio class. The tests will not compile because the Radio has not been implemented yet. (Add reference).

Implement the Radio

Implement the class Radio so that it corresponds to the following class diagram and specification, and passes all the tests. Do not change the tests.

Radio
Class

◢ Fields
- 🔒 _channel : int
- 🔒 _on : bool

◢ Properties
- 🔧 Channel { get; set; } : int

◢ Methods
- ⬦ Play() : string
- ⬦ TurnOff() : void
- ⬦ TurnOn() : void

- The default value of _channel is 1 and _on is false
- The radio has four channels: 1, 2, 3, 4
- The channel can only be changed when the radio is on.
- The radio can play only if it is on, in which case it returns a string indicating which channel it is playing, for example:
  - `"Playing channel 4"`
- If an attempt is made to play the radio while it is off, the string `"Radio is off"` is returned.

Graphical User Interface

Once the Radio class is complete and all the tests pass, use WPF to add a Graphical User interface to the radio.

Remember that the radio logic should be in the Radio class, not the interface code-behind class. Change as little of the Radio code as possible.

Initially the GUI should simply display the strings returned by the radio methods (“Radio is on” or “Radio is off”, “Playing Channel 4”)

Once you have gotten the basic functionality working, add more functionality to the radio and its GUI, for example the ability to increase and decrease the volume. Add unit tests for the new functionality.