Object Oriented Programming

Topic 2: Unit Testing

Lecture Demo - Unit Testing the Spells

Return to your School of Magic program. Add Unit Tests to check the functionality of the Spell class.

- Create one test for each spell kind to check that casting it returns the correct data (3 tests)
  - For example: check that a Heal spell returns "Ahhh… you feel better".
- Create one test to check that you can change the name of a spell

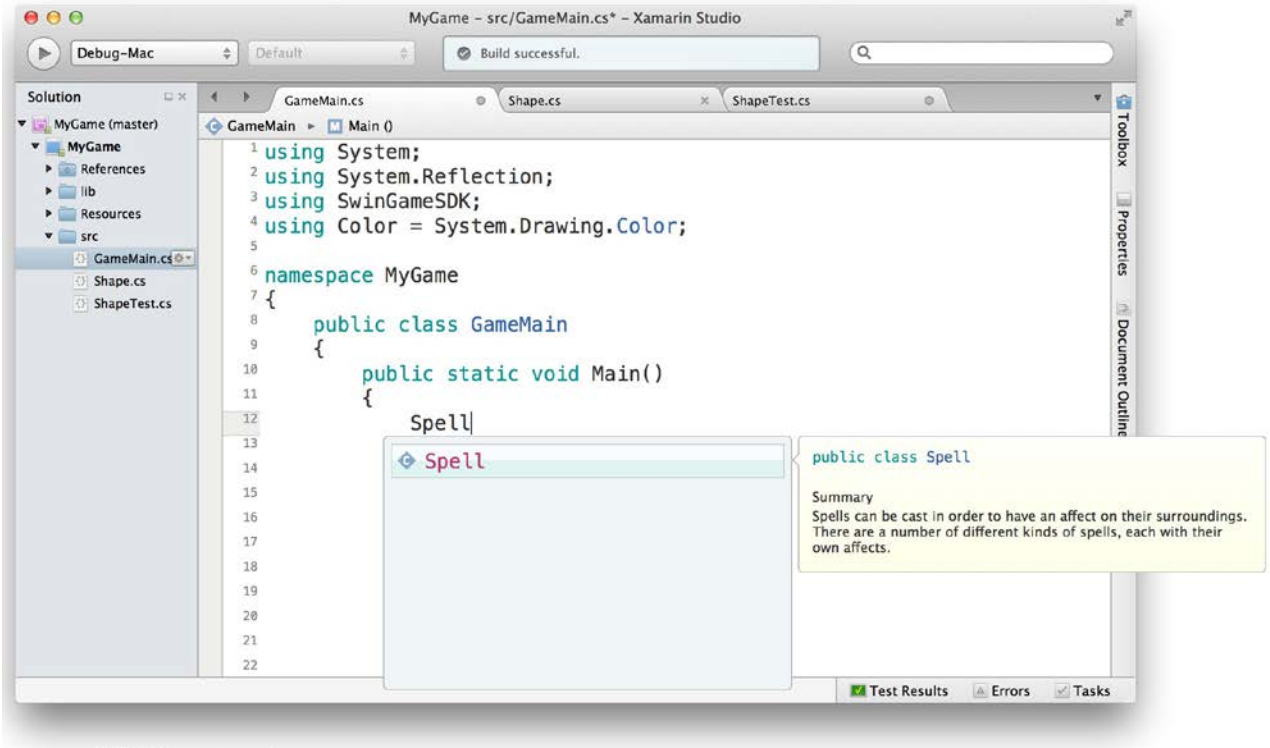
Lecture Demo - Documenting the Spell Class

In a real-world project it is critically important to document you code. Software is developed in teams, and documentation can help everyone quickly understand what the classes, methods, and properties of your code do.

C# provides a code documentation standard called XML documentation. This provides a means of

```
/// <summary>
/// Spells can be cast in order to have an affect
/// on their surroundings. There are a number of
/// different kinds of spells, each with their own
/// affects.
/// </summary>
public class Spell
{
    /// <summary>
    /// Cast this spell, causing it to have an
    /// effect on its surroundings.
    /// </summary>
    /// <returns>a description of the effect</returns>
    public string Cast()
    {
```

The great thing about this documentation is that it is read and understood by the IDE, so when you document the method and classes in your project that documentation is immediately useful.



1. Open your School of Magic program
2. Add XML documentation to the Spell class by adding /// to the line before the class, constructor, method, or property header. This will start the documentation for you, though you can add extra tags in < … > after the summary.
   2.1. Write a short description for the class itself.
   2.2. Write descriptions for the constructor, methods and properties within the class
3. Check that you can see this documentation in the IDE