Object Oriented Programming

Topic 4: Class Inheritance and Polymorphism

Lecture Demo - The Spell Book

Return to your School of Magic program. This week you will change the implementation to use inheritance to model the different kinds of spells.

1. Use the following description to **draw** your own **UML class diagram** of the Spell Book, Spell class and its child classes.

There are three different kinds of spells: Teleport, Heal, and Invisibility. These output different messages when executed.

- Teleport - these spells only work 50% of the time. When they do work the spell outputs the message "Poof… you appear somewhere else", otherwise it outputs "arrr… I'm too tired to move".

> **Hint**: Create a private **static** Random field in the Teleport spell class, and assign it a new Random object. You can then access this from any Teleport object. To get a random value between 0 and 1 you can use NextDouble.
>
> ```
> private static Random _random = new Random();
>
> …
>
> _random.NextDouble()
> ```

- Heal spells always output the message "Ahhh… you feel better".
- Invisibility spells can only be cast once, after that they output the message "pzzzzit".

> **Hint**: Add a boolean field _wasCast to the Invisibility class. Initialise it to false. You can then check and change this in the Cast method.

2. Add new unit tests to the Spell Test class to check the 3 new derived classes; e.g. test that you can cast an Invisibility spell only once.
3. Implement the Invisibility, Heal and Teleport classes.
4. Modify the Spell Book Test class to check that you can add, remove and fetch spells of any type. (Do you need to modify the Spell Book class itself? Why or why not?)
5. Run your tests and make sure they pass.
6. Add XML documentation to your new classes