Object Oriented Programming

Topic 1: Objects and Encapsulation

Lecture Demo - Spells

This task is the first in a small Wizard simulator program that you will create over the next few weeks. In this task you will create a Spell class and a Spell Kind enumeration.

Each Spell will.

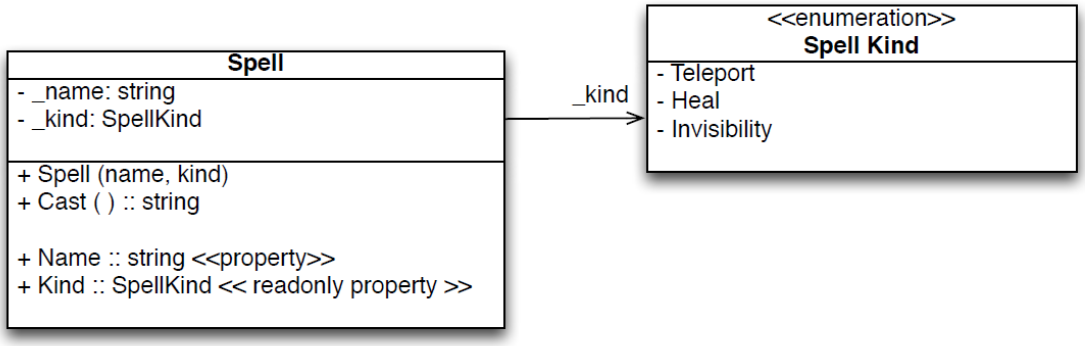
- Know what *kind* of spell it is
- Know its *name*
- Can you **constructed** with a given name and kind.
- Can be **cast**, which returns a message indicating what happened.

There are the following Spell Kinds, and they return the indicated messages:

- Teleport
  - "Poof… you appear somewhere else"
- Heal
  - "Ahhh… you feel better"
- Invisibility
  - "Zippp… where am I?"

You will create a program to test that you can create and work with Spell objects.

The following UML diagram shows the Spell class and the Spell Kind enumeration. The arrow between the two indicates that the Spell **has a** _kind it knows that is a Spell Kind. You can also see this relationship due to the presence of the _kind field.



Note: Normally a UML diagrams would just show the relationship between Spell and Spell Kind and *not* include the field. You would still need to add the field when you wrote the code, but it saves time rather than duplicating this information on the diagram.

1. Create a new **Console Project** called **Swinwarts School of Magic**
2. Add a new file, but this time choose **Empty Enumeration**. Name the file, and its enumeration, **Spell Kind**.
3. Implement the Spell Kind enumeration using the following code. This lists the valid options for this enumeration.

```
public enum SpellKind
{
    Teleport,
    Heal,
    Invisibility
}
```

4. Add a new file for the **Spell** class.
5. Implement the **fields**, and the **constructor**.
6. Add **properties** for Name and Kind.
7. Implement the **Cast** method. It checks the Spell's kind and return the appropriate message.

At this point you should have created all of the code needed for the Spell class. So, now you need to test that it works in the MainClass.

8. Switch to the **Program.cs**
9. Add a **Cast All** static method that calls Cast on all of the elements in an array of spells that is passed to it. It should then use the **Console** class to write the name of the spell, and its effect (result of calling Cast) to the Terminal.
10. Add an array of 5 spells to **Main**.
11. Initialise these with 5 new Spell objects:
    - "Mitch's mighty mover" - Teleport
    - "Paul's potent poultice" - Heal
    - "David's dashing disappearance" - Invisibility
    - "Stan's stunning shifter" - Teleport
    - "Lachlan's lavish longevity" - Heal
12. Use **Cast All** to cast all of the spells.