Class Adapter: Here, there is a class, called TeacherForDADA that implements the TeachingDADA interface. There is also a class called AdapterForSafeTeaching that transforms the spells contained in the TeacherForDADA class so that the students can not use dark magic.

Object Adapter: Here, there is a class, called TeacherForDADA that implements the TeachingDADA interface. There is also a class called AdapterForSafeTeaching that transforms the spells contained in the TeacherForDADA class so that the students can not use dark magic.

Pluggable Adapter: This pattern functions the same as the previous two, except that it allows there to be different "styles of teaching" for different teachers, hence the TeacherNuovoForDADA and TeacherVecchioForDADA classes.

Bridge: Here, the implementation classes of each different type of creature (each extending the Humanoid abstract class) is separated from the class that manages each type of creature (keeps track of how many of each object are registered in a hashmap, sets appearances, etc.) in order to maintain separation between the objects and their implementations and allow for easier "pluggability."

Composite: Here, there is a singleton in the MinisterForMagic that ensures that only one MinisterForMagic object is instantiated at any time. The composite pattern allows classes that implement the Wizard interface to maintain all the functionality of the Wizard interface, while connecting this to the MinisterForMagic class and the classes that extend it.

Decorator: Here, each DeliverMessageThru class is altering the functionality of the original CoreMessageDeliveryClass class. This adds the method of delivery as functionality to the "core class."

Facade: Here, the WizardFacade, UnderageWizardFacade, and WarlockFacade classes are hiding the implementation of each node in the network, providing the user with a less complicated and convoluted experience. Each of these classes hide unnecessary information about their respective object types, ensuring that the user doesn't experience the entire functionality at once.