CS 245
Assign 5: Decorators — Monday Apr 1

Objective:

- Apply the Decorator Pattern to a practical application.

Problem:

Different messaging systems require data to be formatted with different filters before it is sent.

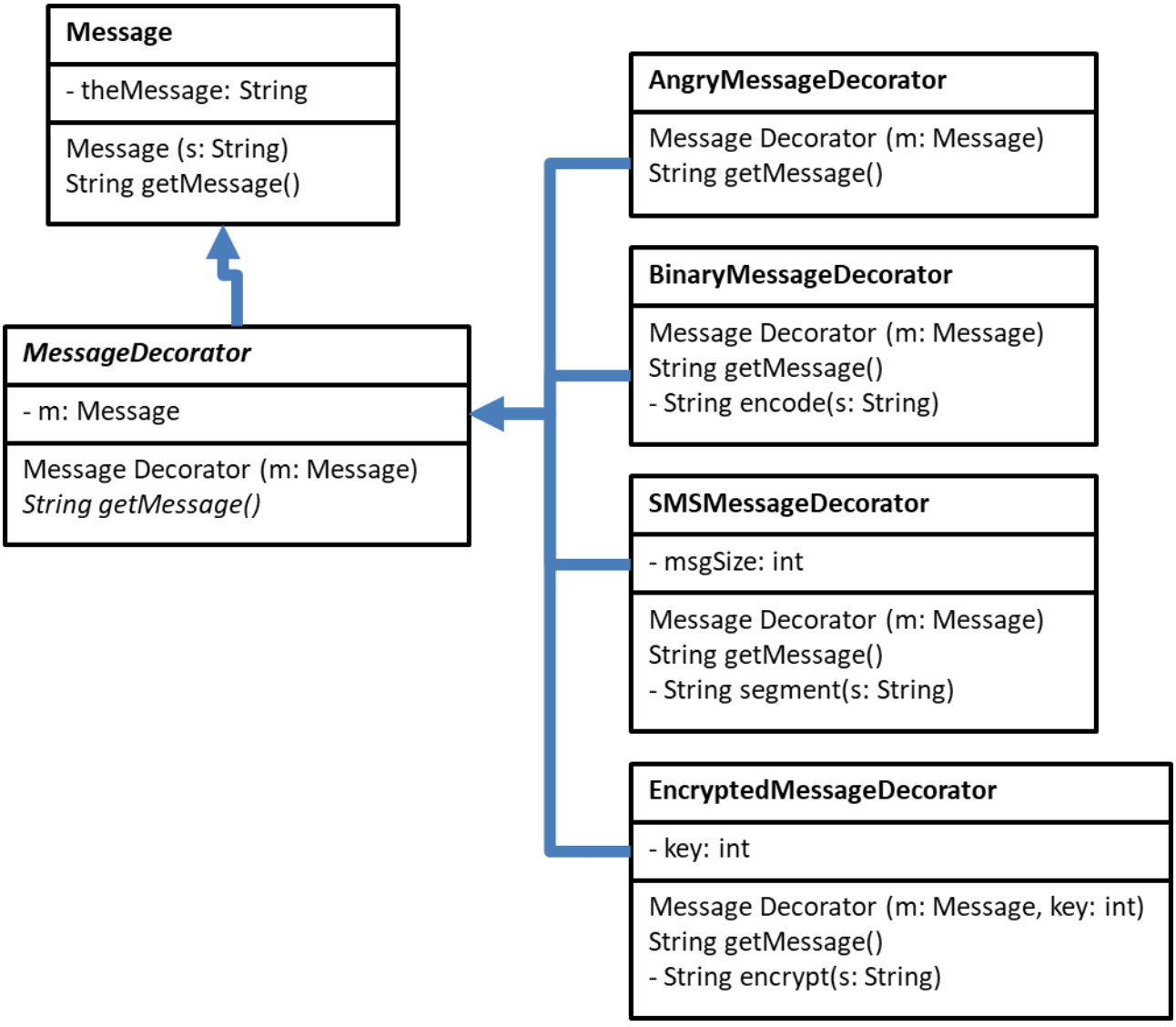
For example, an SMS message is (typically) broken down into blocks of less than 153 characters. Sometimes messages are sent as text (Unicode), sometimes they are formatted as raw Binary. Sometimes messages are encrypted.

We ***could*** create a separate class for each type of message, i.e. EncryptedBinarySMSMessage or UniCodeMessage or …, but that would lead to an explosion of subclasses. Moreover, if we added another filter, we would have to create a whole bunch of additional classes.

This problem lends itself to the Decorator Pattern. Instead of creating separate classes for each type, we can dynamically update the composition of a specific object.

Activities:

1. Implement a simple Message class, an abstract Message Adapter and four concrete decorators: Angry, Binary, SMS and Encrypted.



The decorators should affect the message as follows:

- **Angry**: Converts the message to ALL CAPS
- **Binary**: Converts each letter in the string to its binary equivalent. (hint: Don't reinvent the wheel --- there is probably something in Java that can give you a *binaryString)*
- **SMS**: segments the message into blocks that are no longer than the msgSize. These blocks are still represented by a single string, but they are delimited by the string "…\n". For example if the msgSize is 3 and the message is "Legendary", the result would be "Leg…\nend…\nary"
- **Encrypted**: should use a simple Caesar Cipher to shift the letters of the message by the key. For example: "hal" with a key=1 should produce "ibm". (Note: the point of the activity is not to write an elaborate encryption method – Not even an elaborate Caesar Cipher. This should be able to be accomplished by looking at each character in the string and performing some simple casting and arithmetic).

2. Create a main class/method that can exercise these strategies. This class should
   - Prompt the user for a filename and be able to read the file.
   - Files are anticipated to be of the following format:
     - Line1 contains a message composed of characters in the Ascii range 32-128
     - Line 2:N contains a string that has the name of the decorator; ("Angry", "Binary", "SMS" or "Encrypted")
       - If the text is "Encrypted" it will also be followed by a number for the key (on the same line)
   - The program should apply the collection of decorators to the message (in the order they appear in the file) and print the result to the console.

Your code will be evaluated on functionality ***and*** adherence to good OO design principles