INFO5125

Assignment 2

You will be creating a program that works with social media posts. Posts get "downloaded" from a server, and then can be edited. You can also create new posts.

You will start with a list of posts, which is a list of Proxies. You can enter a command to download a post from the server based on its "ID", which will fill in the proxy. You can then set a post's content or title with a command. Posts are also viewable with a command (i.e. printing them to the console). You can also create new posts.

Details

Post

- A Post consists of a name and body.

Proxy

- Posts start off proxied before they are downloaded.
- Your proxy should **NOT** automatically load posts.

Prototype

- When creating a new post, it will have a default name and title.
- This should clone a Post prototype.

Template Method

- The method to print a post should be a template method.
- Post should have a Print template method which can't be overridden.
- Post should have a GetPrintableTitle and GetPrintableBody function that are overridable.

The commands you should implement are "list", "new", "download:[number]", "settitle:[number]:[new title]", "setbody:[number]:[new title]", "view:[number]", and "quit" (which just exits the program).

- "list" will list in order the contents of your PostProxy list.

    - "settitle:5:Hello" should set the fifth element in your PostProxy list's title to Hello.
- Downloading a post should just fill in the post content. This doesn't have to actually connect to a web server and you can fake it by just directly setting a Proxy's post content.
- A downloaded post should be of class FancyPost. New posts should be of class Post.

FancyPost

- FancyPost is a subclass of Post that prints the post contents in a fancy way.
- How the post is made fancy is up to you. Be creative!
- This should be implemented using the Template Method pattern.

Starter Content

You are provided with a starter project containing a Post object (which needs to be converted to using a proxy) and a basic main method including some input parsing functionality.

You are also provided with an executable demo of the final program. Play around with this to see what functionality should exist in the final project.

You should not need to create any new classes. All the work for this project will be modifying the starter classes.

The starter project was written using .NET Core 3.0. If you want to deviate from this, feel free to create a new project in Visual Studio and copy over the files.