**Workout 2018-10-08 – Proxy Pattern (Part 2) Name: Caden Perdue**

Instructions: Place all your answers in this document. Place this document in the root directory for the program written for question 4. Zip up that directory and submit the zip file as your assignment.

1. Questions about the CD Cover Example
2. Remind us… In the Proxy Pattern, what two classes MUST implement the same interface?

The proxy object and the real object.

1. According to the Java 8 API, ImageIcon has nine constructors. Which one is used in the ImageProxy class? What does that constructor do? (Be more specific than "It creates an ImageIcon object.")

It uses he ImageIcon(URL location) constructor. This constructor has a specified location from which to retrieve the image that is to be displayed when it’s loaded.

1. What does the method paintIcon (page 473) do in the CD Cover program:
2. Before the ImageIcon is instantiated?

Before image is instantiated, a string is displayed which reads “Loading CD cover, please wait” and a new image is created

1. After the ImageIcon is instantiated?’

The icon is painted on the screen

1. Explain when (or how) the String “Loading CD cover, please wait…” is removed from the screen.

Once the new image is loaded, the repaint method is called

1. Questions about the Matchmaking Example
2. Define Dynamic Proxy.

A dynamic proxy, you create a proxy class on the fly that implements one or more interfaces and forwards method invocations to a class that is specified

1. What is the specific purpose for a Protection Proxy?

The protection proxy controls access to an object based on access rights

1. What is the InvocationHandler?

Invocation Handlers implement the behavior of the proxy

1. The textbook is vague about what a Java Bean is. Define it more formally and completely.

A java bean is a convention for a class that has all private properties (accessed by getters and setters), a public no-arg constructor, and is serializable. A Java Bean is just a class that follows those 3 standards

1. What two classes in our example implement the PersonBean? Which is the proxy and which is the “real object”.

PersonBeanImpl and getOwnerProxy implement the PersonBean, the proxy is getOwnerProxy and PersonBeanImpl is the real object

1. Briefly describe the parameters passed to the invoke method (in the InvocationHandler interface).

Proxy is the object we are operating on, method is method being invoked on the proxy, and the args array are the arguments to be passed with the method

1. In real programming setting for our MatchMaking program, how would we make sure that the right proxy is created for each user?

We would have to check each time that the proxy has been correctly applied

1. Compare and contrast: Decorator, Façade, Proxy, and Adapter?

|  |  |  |
| --- | --- | --- |
| Pattern | Similarities with other patterns in this table | Differences with other patterns in this table |
| Decorator | Increase numbers of classes, similar structure to these other patterns  All of these use composition & delegation | Never creates an object  Allows chaining of decorators |
| Façade | Like Adapter, wraps multiple interfaces | Doesn’t operate on one object  Doesn’t add new behavior  Doesn’t have to implement any particular interface |
| Proxy | Like decorator, stays in place of original object and implements interface of original object | Creates new objects  Doesn’t convert interface |
| Adapter | Like façade, forwards requests to a different interface | Doesn’t operate on one object  Converts an interface |

1. For question 4, see the associated program description document. You are to follow instructions that document and include your answers below (in the workout document here).