**Reading Assignment 8 – Chapter 6 Questions**

|  |  |
| --- | --- |
| Name: |  |

Instructions:

* Read pages 191-223 of your text.
* Type answers to the questions below (don’t remove questions).
* Submit on Blazeview.

1. On page 195, Mary says, “...we don’t want the remote to have to know the specifics of the vendor classes.” Discuss a design principle that should be used to meet this requirement. Why is it important in this case?
2. (Omit) Re-read Joe’s definition of a “command object” 4 or 5 times. Now, restate it in your own language?
3. (Omit) Draw a class diagram, as complete as possible from the analogy on page 199. Feel free to draw on your complete understanding of the chapter.
4. Consider the Objectville Diner analogy on page 199. Now consider the command pattern diagram on page 201. Substitute the appropriate class name for each of the names on the diagram (*e.g.* who is the invoker in the diner? command, invoker, receiver)
5. Consider the Remote Control problem in the text as described through page 205. Now consider the command pattern diagram on page 201. Substitute the appropriate class name for each of the names on the diagram (*e.g.* what is the client in the remote control problem? command, invoker, receiver)
6. The command pattern definition says, “...letting you parameterize other objects...” In the lingo of the command pattern, what is/are these “other objects”?
7. (Omit) Study the UML on page 207 carefully. Assess each association or lack of association. The UML is correct; however, can you question the validity of any of the associations or lack of associations? (I see several things I would change). Explain why.
8. Describe how a factory pattern could be used with the command pattern.
9. (Omit)Suppose there was a pool of equivalent *Receiver* objects and they need to be assigned just as a command was about to be executed (e.g. a load-balancer). How could you modify this classic command pattern situation? Would it still be the command pattern?
10. (Omit) What is the *null object* design pattern? Why is it useful?
11. Consider undo as it is described on pages 216-219. Describe in your own words how *undo* is implemented.
12. (Omit)Draw a class diagram of the Remote example that supports Undo. Show methods. You only need to show a few commands.
13. Why is state information sometimes needed when implementing *undo*?
14. Describe how the *macro command* is implemented?
15. (Omit) How do you implement multiple undo’s?
16. (Omit) Briefly describe in your own words, the use of the command pattern in *queuing requests*.
17. (Omit) Briefly describe in your own words, the use of the command pattern in *logging requests*.

The text says that we need to add two methods for each Command: store and load. Presumably, when a Command is executed, it is also stored. Not sure how this works, exactly.