Assignment: Design Patterns

**Objectives:**

* Reinforce the method used in class to learn design patterns
* Reinforce understanding of class/package diagrams
* Practice using design patterns to solve design problems

**Part I: The Learning Method** (40 pts)

In class, we have used the same method to learn five design patterns. Please summarize what you have learned by filling out the table below.

|  |  |  |
| --- | --- | --- |
| Design Patterns | Problem to be solved | Solution |
| Iterator |  |  |
| Composite |  |  |
| Singleton |  |  |
| Observer |  |  |
| Strategy |  |  |

**Part II: Class/Package Diagram** (10 pts)

Please read the following diagram carefully, and answer Question 2.1 – 2.3.



2.1 How many packages are in the diagram? What are they?

2.2 What do  and  represent respectively? What is the difference between them?

2.3 Why methods in an interface are *italicized*?

**Part III: Application** (50 pts)

3.1 (10 points) Try to understand the following program.

|  |
| --- |
| import java.util.Iterator;  import java.util.LinkedList;  public class IteratorDemo {  public static void main(String[] args) {  LinkedList<String> cities = new LinkedList<String>();  cities.add("Chicago");  cities.add("Denver");  cities.add("Miami");  cities.add("Los Angeles");  cities.add("Seattle");    Iterator<String> iterator1 = cities.iterator();  Iterator<String> iterator2 = cities.iterator();  System.out.println("Iterator1 type for the datastructure is: " + iterator1.toString());  System.out.println("Iterator2 type for the datastructure is: " + iterator2.toString());  while (iterator1.hasNext()){  String city1 = iterator1.next();  String city2 = iterator2.next();  System.out.println(city1+", "+city2);  }  }  } |

3.1.1 What is the expected output (based on your understanding of the code)?

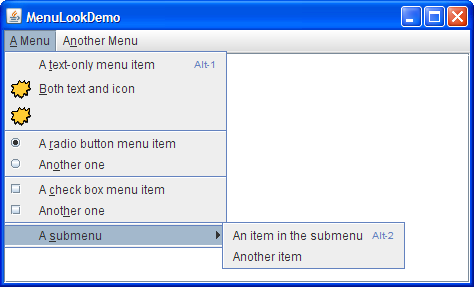
3.1.2 Run the program in Eclipse, and compare the actual output with your answer in 3.1.1.

3.1.3 What did you learn from this example? You may visit the following link for more insights:

<https://sourcemaking.com/design_patterns/iterator/java/1>

3.1.4 Modify the program so that a Hashset is used instead of a Linkedlist. Which line(s) should be modified?

3.2 (10 points) In Java Swing, menus are constructed using the composite pattern.

Briefly explain how the composite pattern is used to construct the above menu.

3.3 (10 points) In Windows 10, only one instance of the “Task Manager” object is needed. Design and implement a class called “TaskManager” using the singleton pattern. (Do not need to consider thread safety.)

3.4 (10 points) Consider a simple bidding system which has the following functionalities:

* Display the latest bid to online bidders
* Announce the latest bid to call-in bidders (who are on the phone)
* Save all bids to a database.

Use the observer pattern to design this system, and present your design using a class diagram.

3.5 (10 points) Design a program that can sort an array using different sorting algorithms, such as quick sort, merge sort, bubble sort, insertion sort. Present your design using a class diagram.