Tambola/ Housie Project

Documentation

By:-

Sarthak Tayal- 2020A7PS0987P

• Encapsulate what Varies

Encapsulation is followed, like in Tambhola class finishedNumbers, randomGenerator , etc. , in Ticket class TicketId , String Numbers follow encapsulation because they are private thus they can only be accessed using class functions.

• Favour composition over Inheritance

Yes, this is followed. The Thambola class contains a TicketGenerator object as its data member, thus there is no inheritance used but Thambola contains TicketGenerator in its composition.

• Classes should be open for extension and closed for modification

Classes are not very freely open for extension but they are closed for modification. It is possible to freely add features but it
will require certain amount of change in the code.

• Depend on abstraction, do not depend on concrete classes

No, since we are creating objects of the classes in the program , we require concrete classes. There is no abstraction as all the classes used in the program have their objects created somewhere or another.

Design Pattern

Code is resembling to Observer Pattern.

The **observer pattern is a software** design pattern in which an object, named the **subject**, maintains a list of its dependents, called **observers**, and notifies them automatically of any state changes, usually by calling one of their methods.

Observer Design Pattern is used as the skeleton of the application. In this case, as per the observer pattern, moderator is the subject and players are the observers. The change in state is generation of a new number and update by the observers is checking the number and striking it off, if present