Carlos Guisao

Homework 4

a) What is the purpose of a design pattern?

The purpose of a design pattern is to provide general solutions to design problems that occur frequently in OOD.

b) When do you apply the Observer pattern?

The observer pattern should be used when a change of a state in one object must be reflected in another object without keeping the objects tightly coupled. While leaving the door open to future enhancements like adding more observers with minimal change.

c) You review a design written by somebody else for an application and you find these:

- an interface Shape with a method draw()

- a class Circle that implements Shape

- a class Rectangle that implements Shape

- a class CompoundShape that:

o implements interface Shape

o aggregates 0 or more Shape objects,

o has an extra method called add(Shape sh)

o for implementing method draw() calls the draw() method for all

aggregated Shape objects.

You assume that a CompoundShape object is made of multiple shapes.

What design pattern is at work in this application? Explain your answer.

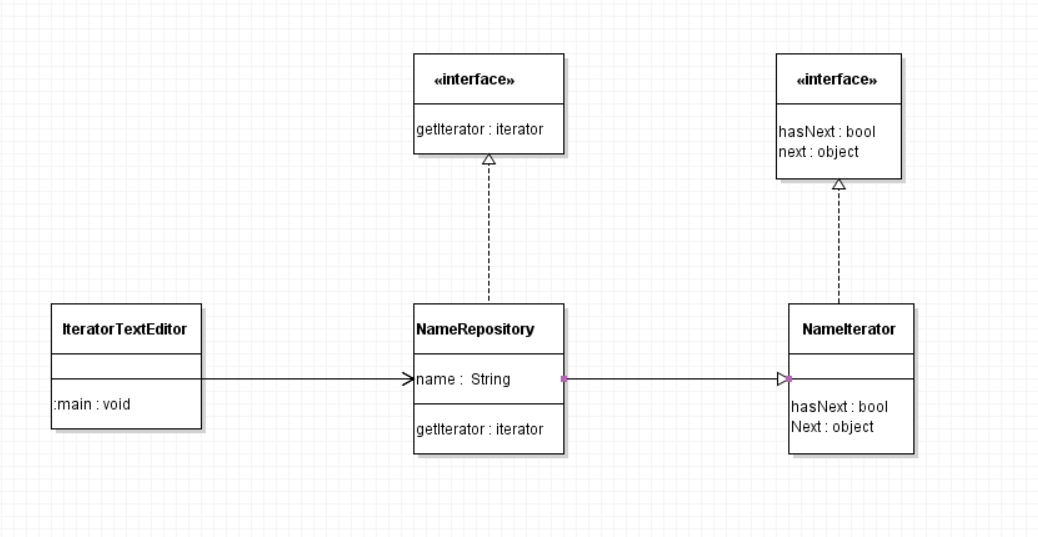
The composite design pattern is being used here in this example. The intent of a composite pattern is to group components into a whole which is what the CompoundShape class is trying to achieve. Example add(Shape sh) method.

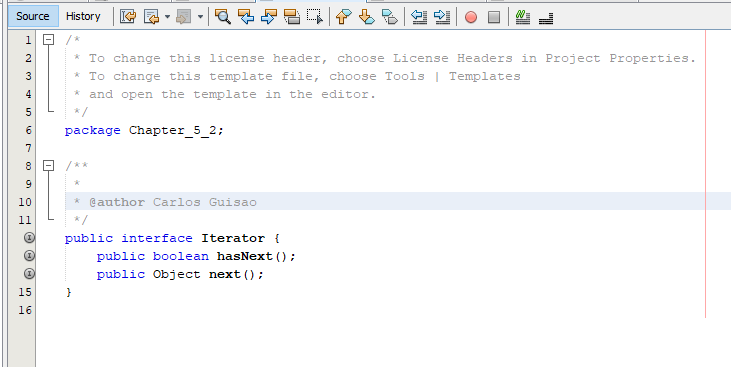
d) The TitledBorder class can give a title to a border. Consider the code

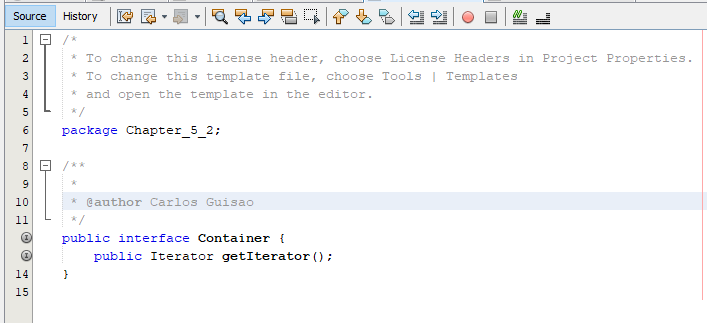
The pattern at work here in the strategy pattern because the method is being invoked with instances of other classes without declaration. In this example the context class Title Border calls the selected method of the strategy object EtchedBorder.

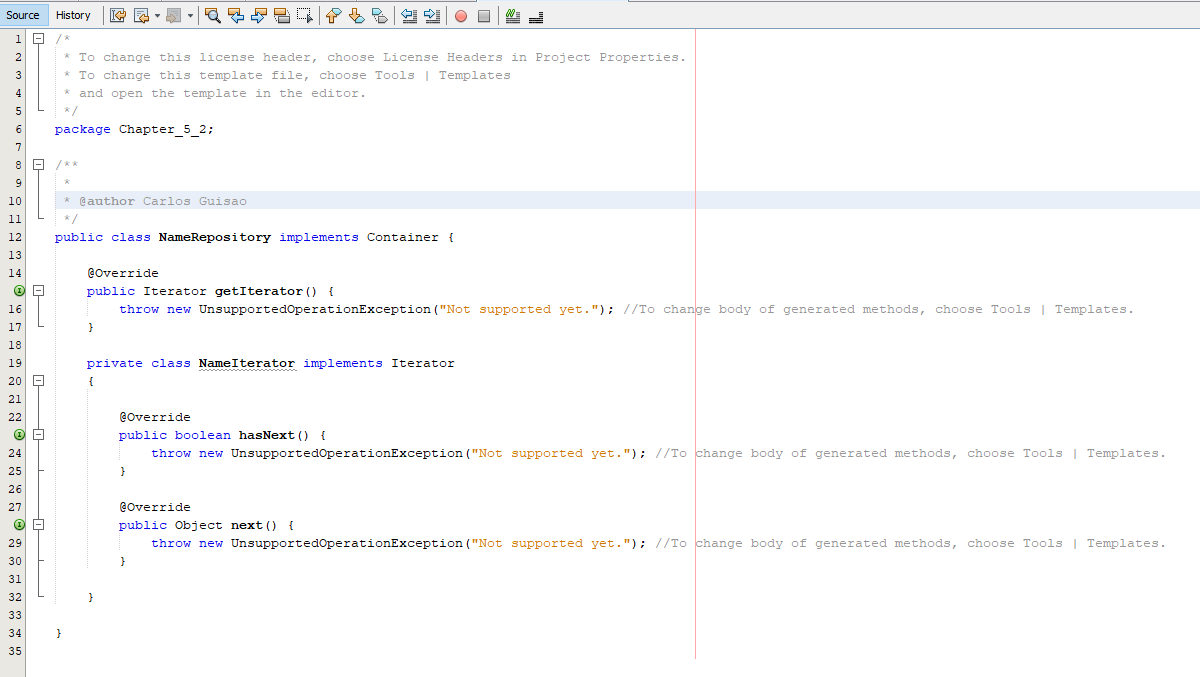
5.2.

a) I would use an iterator pattern because this example requires different versions of a spell-checking algorithm to be considered. This pattern makes the most sense for code that is maintainable over a period.

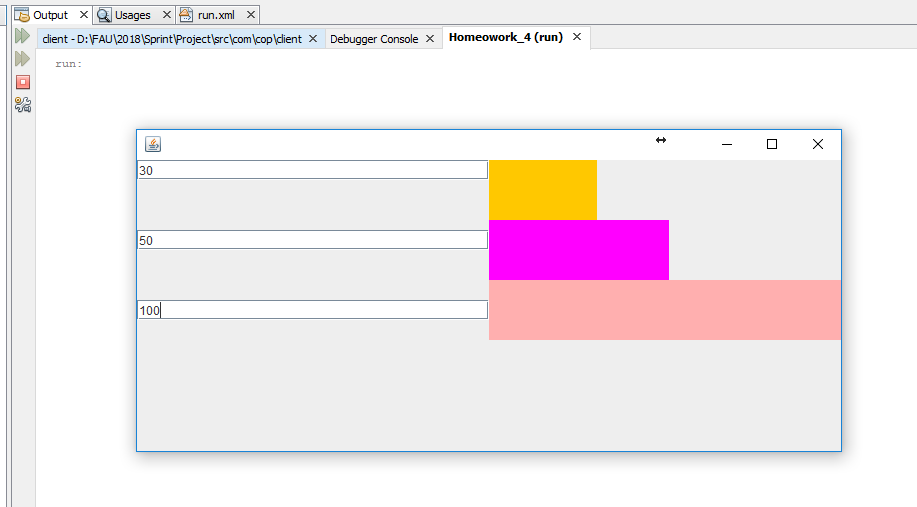
b) 

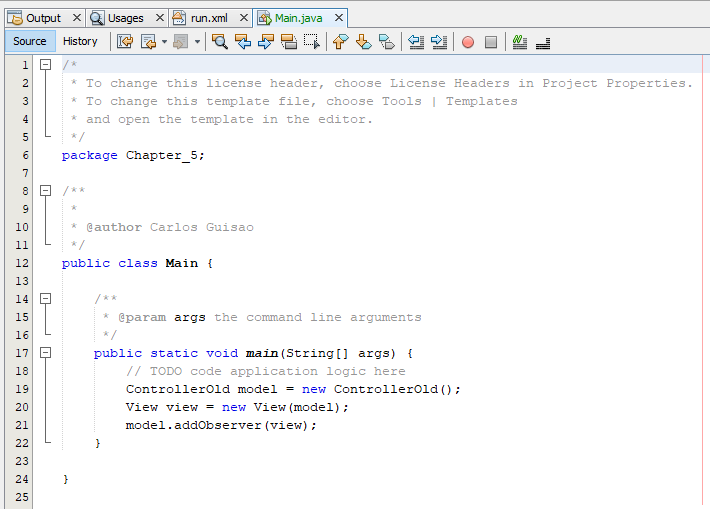
d) 

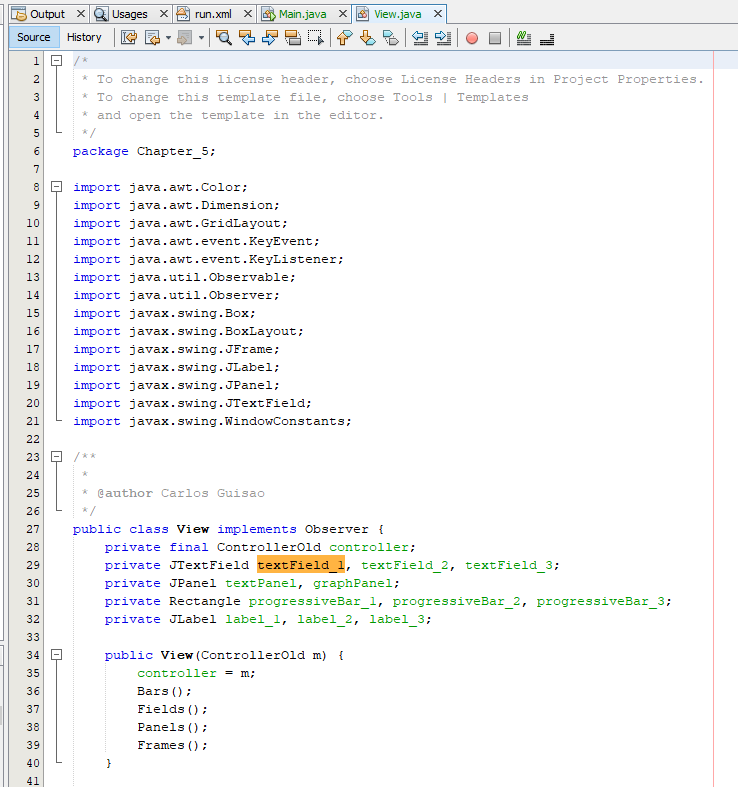


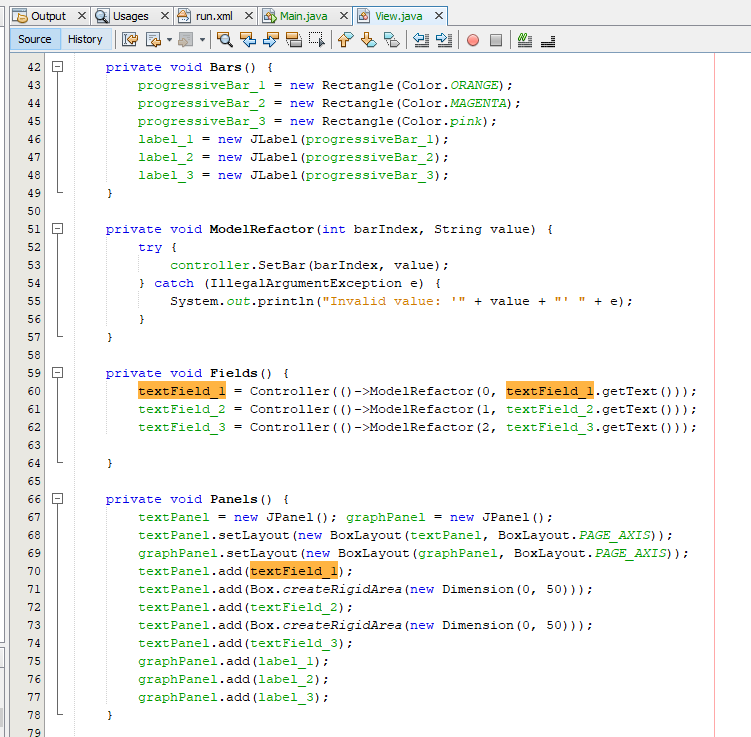


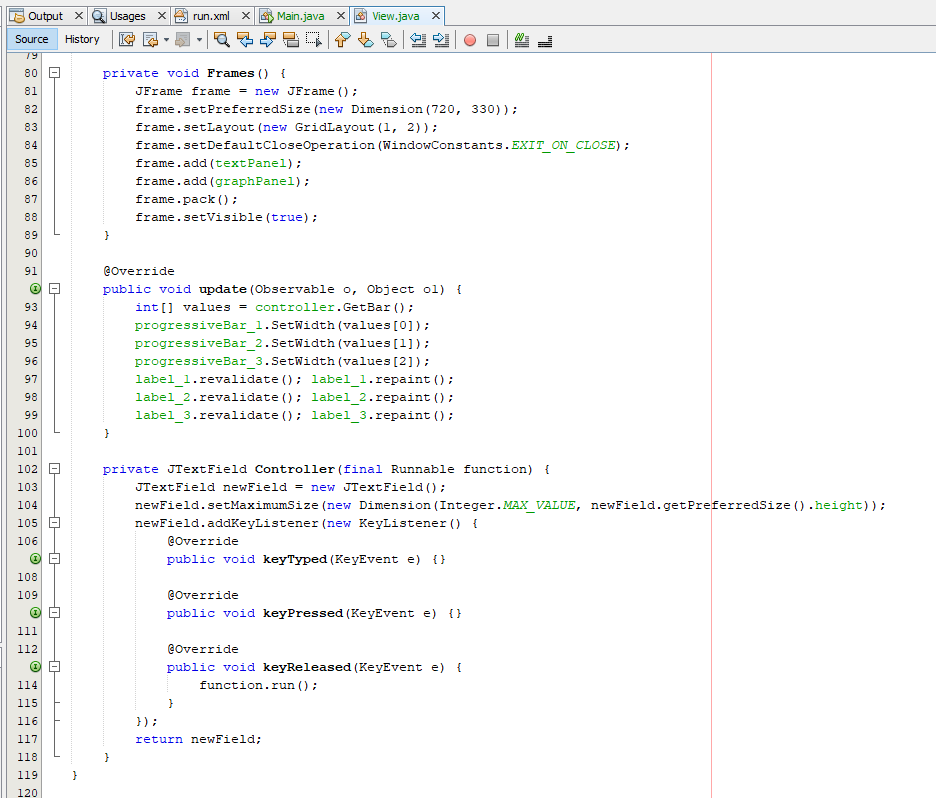
5.3











Chapter 6

6.1

a) Explain the purpose of abstract classes in no more than 15 lines.

The purpose of an abstract class is to define the basic functionality while leaving certain methods or parts undefined. By doing this abstract classes function as the base class. All undefined areas are left up to the person extending the class to define. Abstract classes are also used in situations where you want to design a class but do not want to allow anyone to make an object of that class. By making a class abstract you achieve this feature.

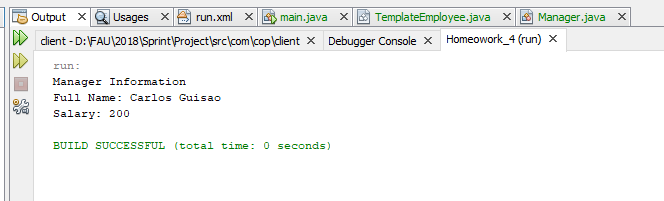
b) Give an example for a situation when an abstract class cannot be used in a Java program and an interface is the only choice.

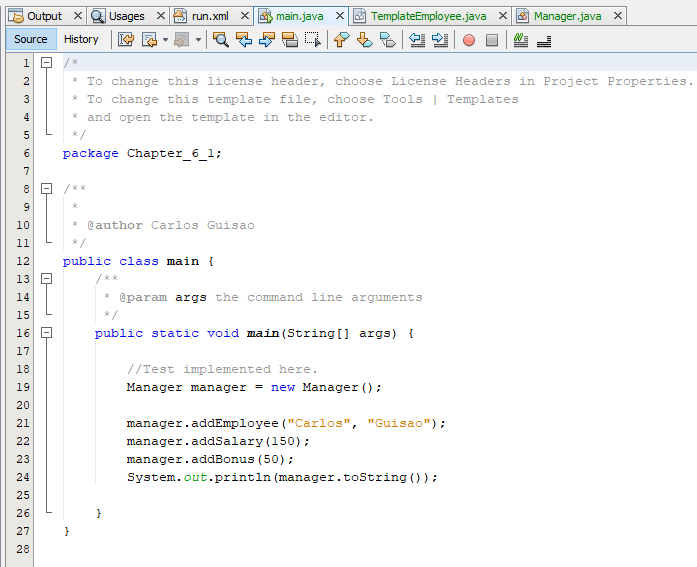
An example of when you cannot use an abstract class in a java program is when you want to take advantage of multiple inheritance type. Like the HashMap class in JDK that implements several interfaces like Serializable and Cloneable.

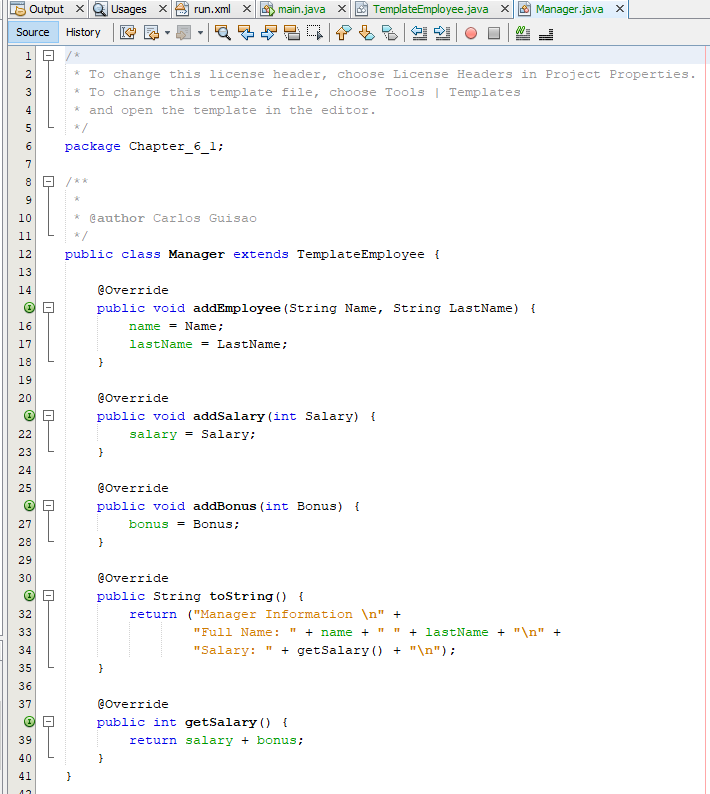
c) GeneralPath collects shapes and is itself a shape. What design pattern does it implement? Explain.

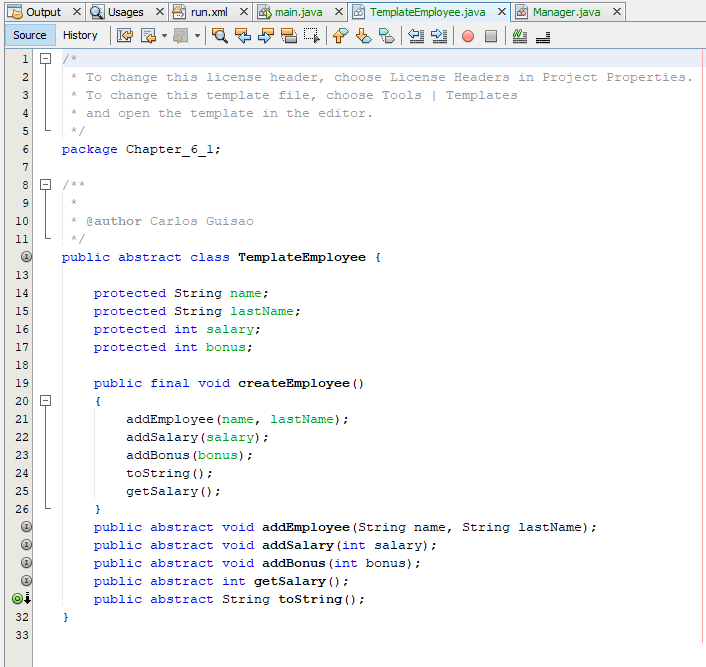
The design pattern here at work is a composite pattern. GernalPath collects shapes and is also a shape this is a component/container problem discussed in class. Therefore, the pattern working here is the composite pattern.

6.2









6.2

