quiz solution w2

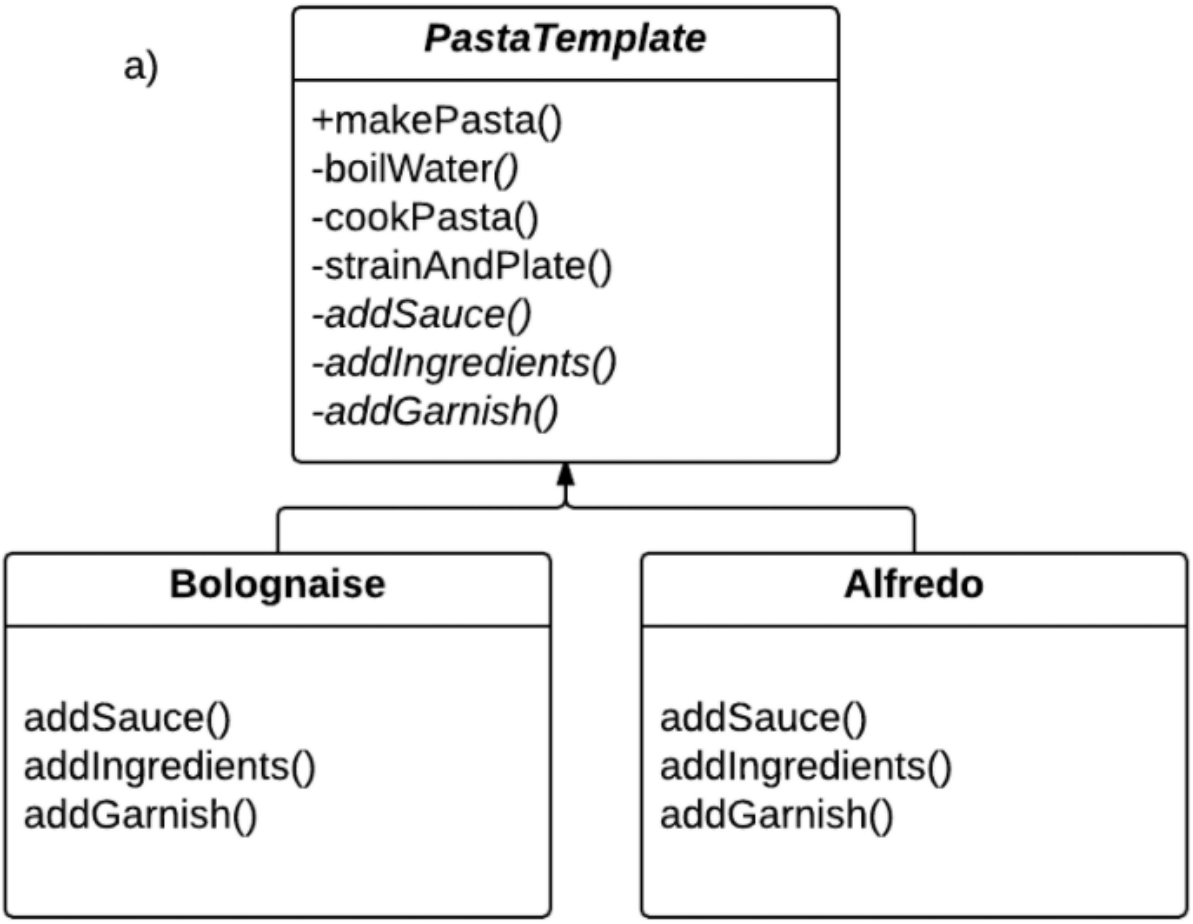
Question 1

Choose the most appropriately implemented Template pattern.

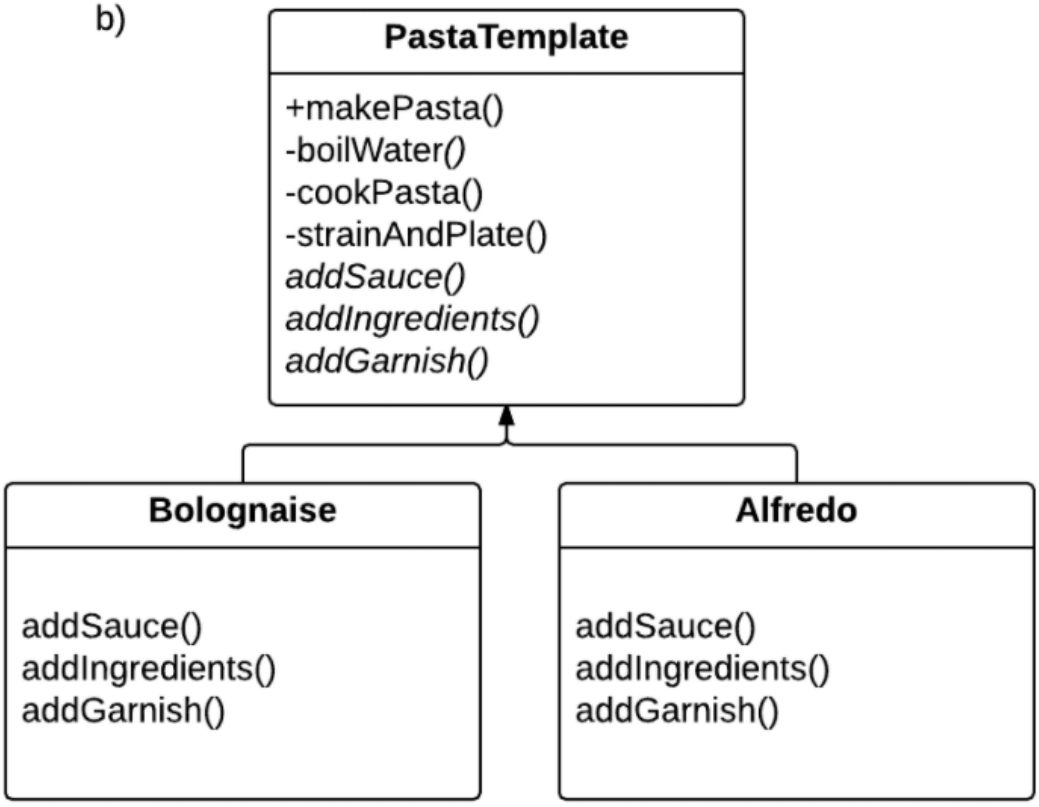
Some UML reminders that will help you:

1. a private method or variable is denoted by a - as in -boilWater().
2. a method, variable, or class that is abstract is denoted by italics (as in PastaTemplate)

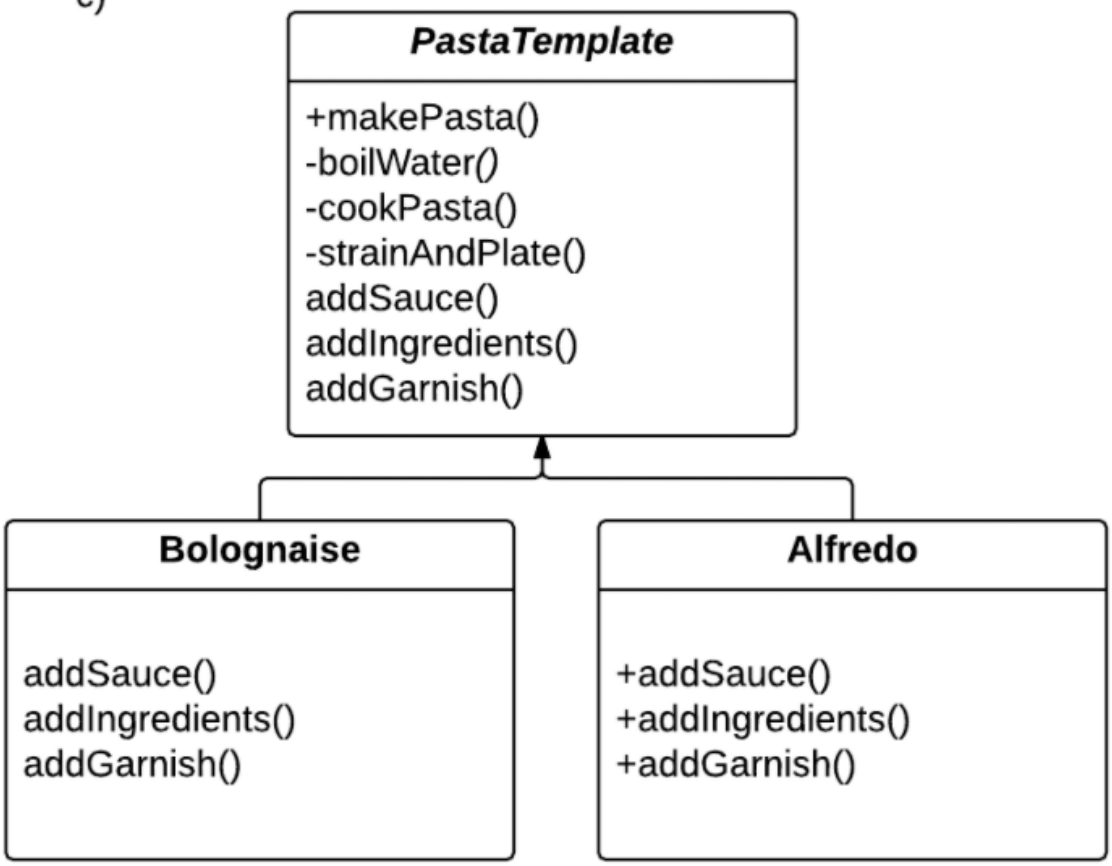
a)



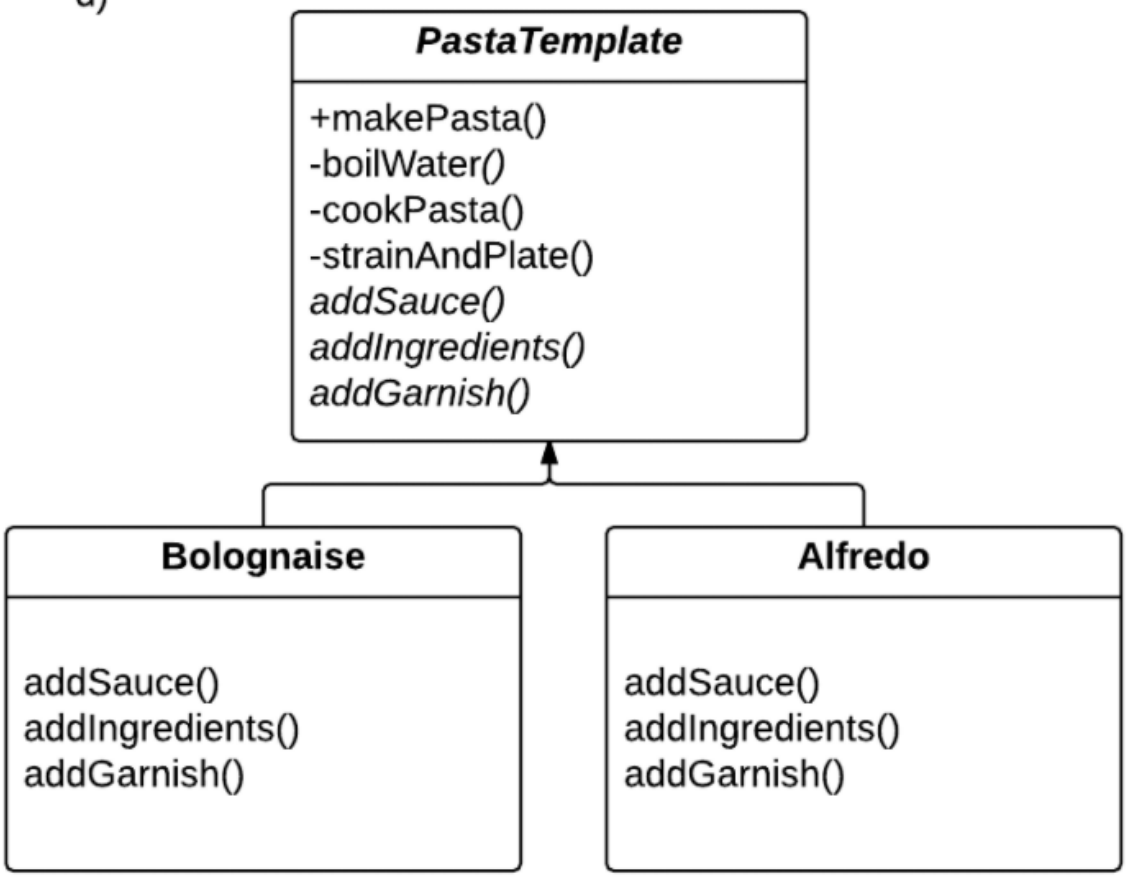
b)


PastaTemplate
+makePasta()
-boilWater()
-cookPasta()
-strainAndPlate()
addSauce()
addIngredients()
addGarnish()
Bolognaise
addSauce()
addIngredients()
addGarnish()
Alfredo
addSauce()
addIngredients()
addGarnish()

c)


PastaTemplate
+makePasta()
-boilWater()
-cookPasta()
-strainAndPlate()
addSauce()
addIngredients()
addGarnish()
Bolognaise
addSauce()
addIngredients()
addGarnish()
Alfredo
+addSauce()
+addIngredients()
+addGarnish()

d)



0 / 1 point

☐ **a)**

Incorrect

Incorrect. It declares private abstract methods. This is a contradiction!

☐ **b)**

Incorrect

Incorrect. The superclass is not abstract. If the PastaTemplate is instantiated directly, there will be errors if you try to use the makePasta method.

☐ **c)**

Incorrect

Incorrect. This is not the usual implementation of Template method because the methods that makePasta uses are not abstract. Some of your methods could have default behaviour, however, so this could be used in some cases.

☑ ~~d)~~

Correct

Correct! The key aspect of the template pattern is that some parts of the Template are defined by the subclasses. These are the addSauce(), addIngredient() and addGarnish() methods, and they are in italics because they are abstract in the superclass.

Question 2

What is the correct situation for the use of a Chain of Responsibility pattern?

☐ **You need a set of objects to each contribute information on responding to a request.**

Incorrect

Incorrect. The classes do not collaborate directly.

☑ ~~You have multiple potential handlers, but only one will deal with the request.~~

Correct

Correct! The handlers pass the message down until one can handle it or the end of the chain is reached.

☐ **You need to pass a message to multiple receivers.**

Incorrect

Incorrect. The message goes to multiple receivers only if prior receivers cannot handle it.

☐ **You need to delegate a set of tasks to a hierarchy of objects.**

Incorrect

Incorrect. Think of how the request is handled in Chain of Responsiblity.

Question 3

What is the purpose of encapsulating state in an object in the State Pattern? Choose the **three** that are correct**.**

- [x] ~~it allows the current state object to decide how to achieve behaviours specific to the state of the context.~~

Correct

Correct! The subclasses of state provide that actual implementation of the behaviours.

- [ ] **it allows the current state to be copied from one instance to another**

This should not be selected

Incorrect. Copying state is not the purpose of the State pattern.

- [x] ~~it removes large conditionals that are difficult to maintain.~~

Correct

Correct! The state pattern outsources those "ifs" to a State object - a subclass of State - which decides how to handle requests.

- [x] ~~it turns the context into a client of the state.~~

Correct

Correct! This allows the context to easily make requests of the state.

Question 4

What design principles is the Command Pattern using?

0 / 1 point

- [ ] **Generalization, information hiding, loose coupling**

Incorrect

Incorrect. One of these isn't quite right, try again!

- [ ] **Encapsulation, generalization, information hiding**

Incorrect

Incorrect. One of these isn't quite right, try again!

- [x] ~~Encapsulation, generalization, loose coupling~~

Correct

Correct! The command pattern encapsulates a request as an object, provides a general command interface for managing command objects, and allows you to have looser coupling between the participants.

- [ ] **Encapsulation, information hiding, loose coupling**

Incorrect

Incorrect. One of these isn't quite right, try again!

Question 5

Which are the minimum requirements of the Observer pattern? Choose the **three** that are correct.

0 / 1 point

- [x] ~~update method in observers~~

Correct

Correct! When the observers are informed that a change has been made, they update themselves accordingly.

- [x] ~~method to notify observers~~

Correct

Correct! The subject has a method for notifying the observers that a change has been made

- [x] ~~methods to add or remove observers~~

Correct

Correct! There must be a way to track which observers are associated with a subject.

- [ ] **a state variable to determine if observers have been notified.**

This should not be selected

Incorrect. There can be a flag for determining if a change has been made, but it is not strictly necessary.

Question 6

When are you most likely to need a Mediator pattern?

- [ ] **When your class is sending a request that might be handled by one of several handlers.**

Incorrect

Incorrect. This is a job for Chain of Responsibility.

- [x] ~~**When you are coordinating the activities of a set of related classes.**~~

Correct

Correct! Use a Mediator pattern to coordinate the activities of many, relatively simple classes

- [ ] **When you have two classes with different interfaces that you must connect.**

Incorrect

Incorrect. This could be a job for the Adapter pattern!

- [ ] **When you want to de-couple a class that is requesting a service from one that is providing it.**

Incorrect

Incorrect. This is probably a Command situation.

Question 7

Marlon is coding part of the software that follows a similar sequence of steps. Depending on the type of object, these steps will be implemented in slightly different ways, but their order is always the same. Which design pattern could Marlon use?

- [x] ~~**Template pattern**~~

Correct

Correct! The Template pattern specifies a general 'recipe' and some common steps in the superclass, but allows the implementation of many of the steps in that recipe to the subclasses.

☐ **Mediator pattern**

Incorrect

Incorrect. The Mediator pattern creates an object for managing the interactions between many different objects.

☐ **Command pattern**

Incorrect

Incorrect. The Command pattern turns a command into an object.

☐ **State pattern**

Incorrect

Incorrect. The State pattern also delegates tasks to subclasses, but these have to do with the State rather than an algorithm with slightly different implementations.

Question 8

What are the important roles in the Command Pattern?

☑ ~~**Command, Receiver, Invoker**~~

Correct

Correct! These are the three roles that must be implemented in a Command pattern.

☐ **Delegate, Command, Requester**

Incorrect

Incorrect. Delegate is not one of the command roles, and neither is requester!

☐ **Sender, Receiver, Invoker**

Incorrect

Incorrect. Sender is not one of the command roles.

☐ **Command, Queue, Receiver**

Incorrect

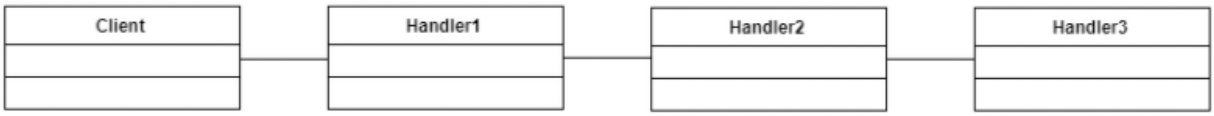
Incorrect. Queue is not one of the command roles.

9.

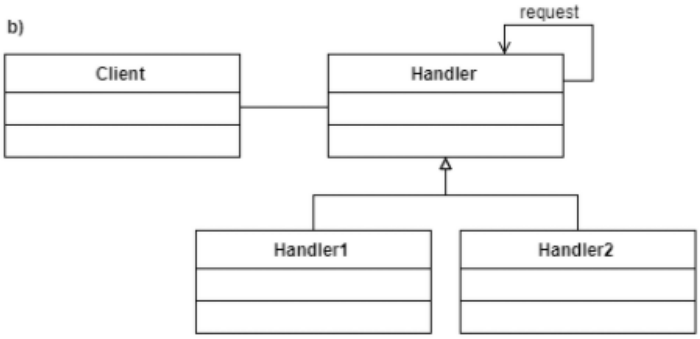
Question 9

Select the best UML class diagram representation of the Chain of Responsibility pattern.

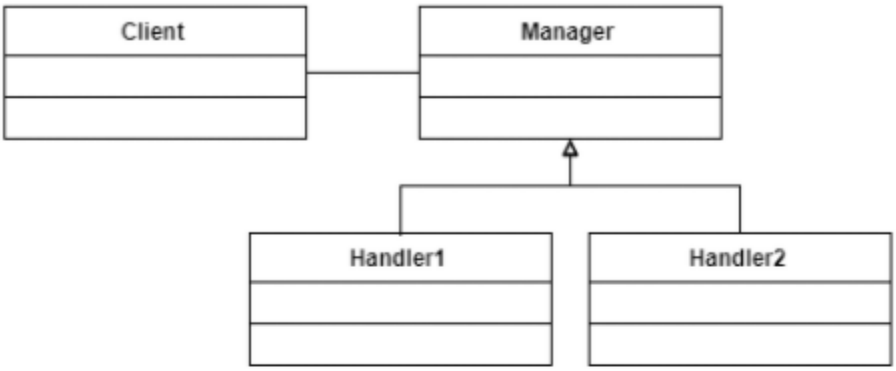
a)



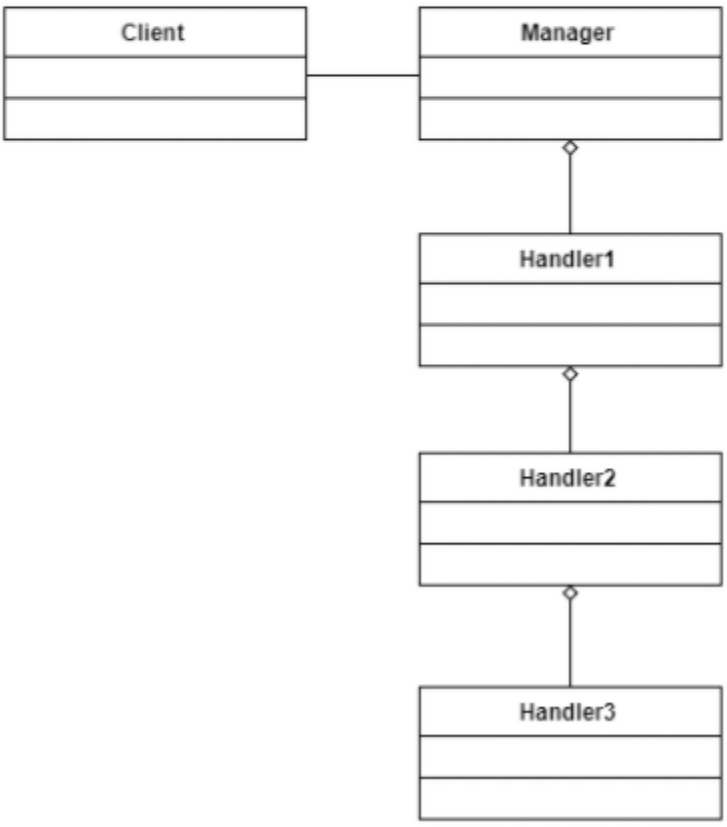
b)



c)



d)



☐ a

Incorrect

Incorrect. The client should not have knowledge of a specific handler.

☑ ~~b~~

Correct

Correct! The Handler superclass manages the logic for determining where to send the request next. Each subclass tried to handle the request.

☐ c

Incorrect

Incorrect. This is close to the proper implementation but usually instead of having a class that manages handlers, it is simply managed in the superclass.

☐ d

Question 10

You have a machine performing a complex manufacturing task, with different sensors and different components of the machine represented by different classes. Which design pattern will you use to arrange the parts?

☐ **Template**

Incorrect

Incorrect. Template is usually used when there are different processes that use many of the same steps.

☐ **Command**

Incorrect

Incorrect. Command is used to handle requests, but its main intent is not in coordinating the activities of many objects.

☑ ~~**Mediator**~~

Correct

Correct! The Mediator pattern is useful when coordinating the activities of many interrelated classes.

☐ **Chain of Responsibility**

Incorrect

Incorrect. The Chain of Responsibility pattern is used when there is more than one potential handler of a request.

Question 11

You have a security system class, and it has 3 modes: normal, lockdown, and open. Which pattern would you use to model the behaviour in these different modes?

☐ **Template**

Incorrect

Incorrect. Template is usually used when there are different processes that use many of the same steps.

☑ ~~**State**~~

Correct

Correct! This pattern is useful when a class has a collection of behaviours that are a little bit different while the system is in different states.

☐ **Observer**

Incorrect

Incorrect. Observer is used when one class needs to monitor the data in another class.

☐ **Mediator**

Incorrect

Incorrect. A mediator pattern is for coordinate the activities of many objects.

Question 12

One of your classes represents a mailbox, while another is the owner of the mailbox. The person would like to know when new mail arrives. Which design pattern will you probably use?

☐ **Mediator**

Incorrect

Incorrect. Mediator is for coordinating the activities of many objects.

☐ **Command**

Incorrect

Incorrect. The command pattern allows you to send a request as an object. Try again!

☑ ~~**Observer**~~

Correct

Correct! Observer pattern is like subscribing. The Owner is alerted when the mailbox has new mail.

☐ **State**

Incorrect

Incorrect. The state pattern allows you to change behaviour based on the state.