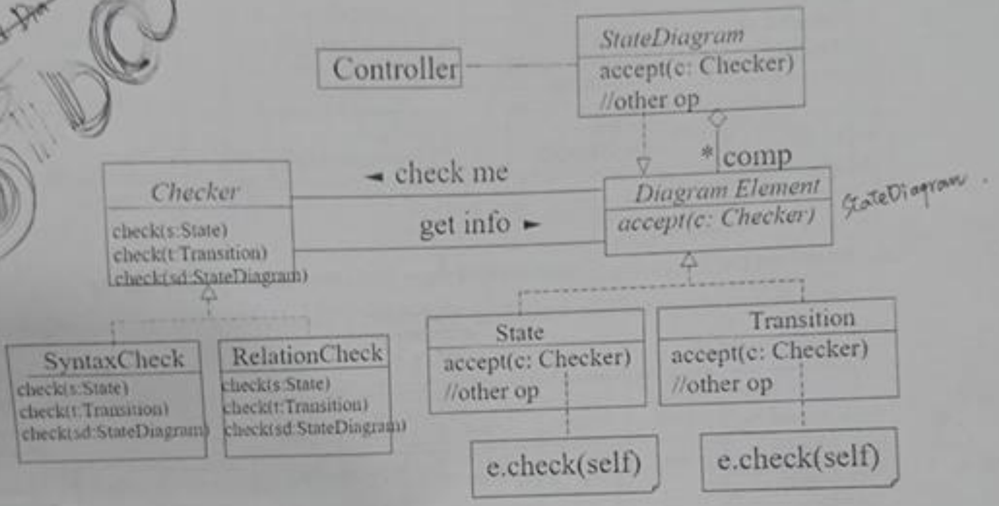
四資管四 B10223007

OOSE Quiz2 5/5/2017 林宜萱 Hsuan

1. The state diagram editor provides analysis capabilities such as checking the syntax and relation of state and transition specifications. We may also add additional operations later on. The problem is that the classes could be overloaded with unrelated responsibilities. An application of the visitor pattern is suggested to decouple these type-dependent operations from the classes of the structure. Please write the visitor pattern in Java code based on the provided diagram. 30%

~~public void Dia~~



2. Suppose the state diagram editor allows the user to add functions of draw scroller or border to the state diagram. The following figure illustrates how the decorator pattern accomplishes this. First, the client can create states, transitions, and the state diagram, and then decorates the scroller or border to the state diagram composite. During repaint, the client calls the draw (g) method of the state diagram. The state diagram called the draw (g) method of the Decorator. The Decorator calls the draw (g) method of the state diagram and then calls its own drawScroller (g) or drawBorder (g) method. In this way, the state diagram with a Scroller or Border is painted. Please write the Java code of the decorate pattern applied in this example. 20%

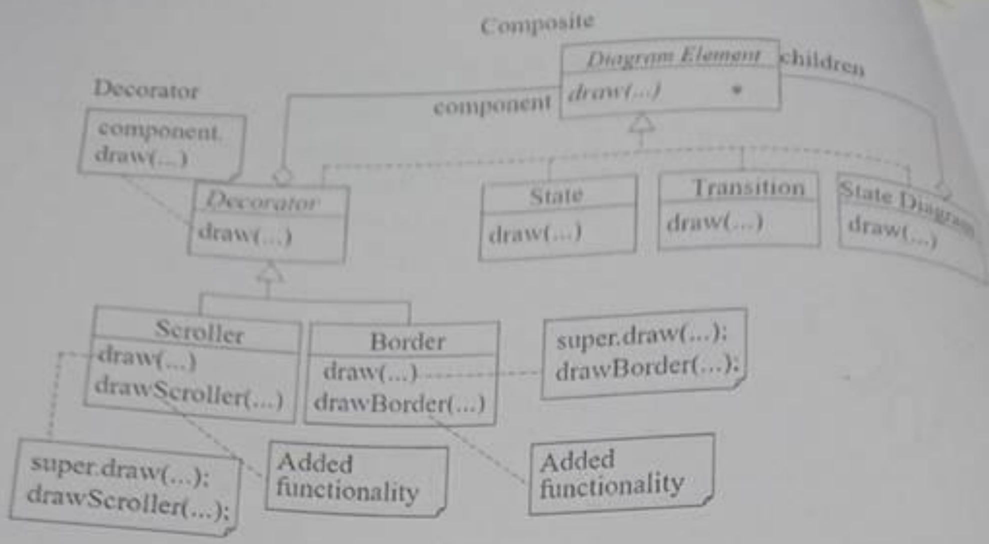
drawScr

Client $\xrightarrow{draw}$ StateDiagram $\xrightarrow{draw}$ Decrator

$\xleftarrow{draw}$

drawBo

String [args]



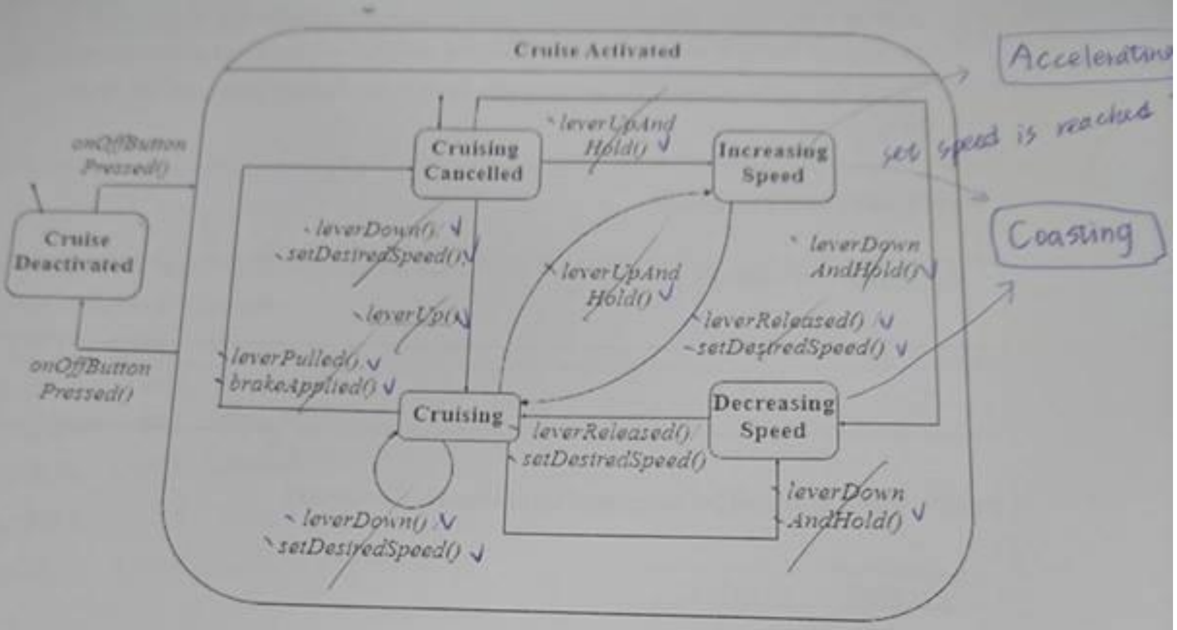
3. Based on the following Cruise control state transition table, please create examples of abnormalities including dead state, unreachable state, and impossible transition. 10%

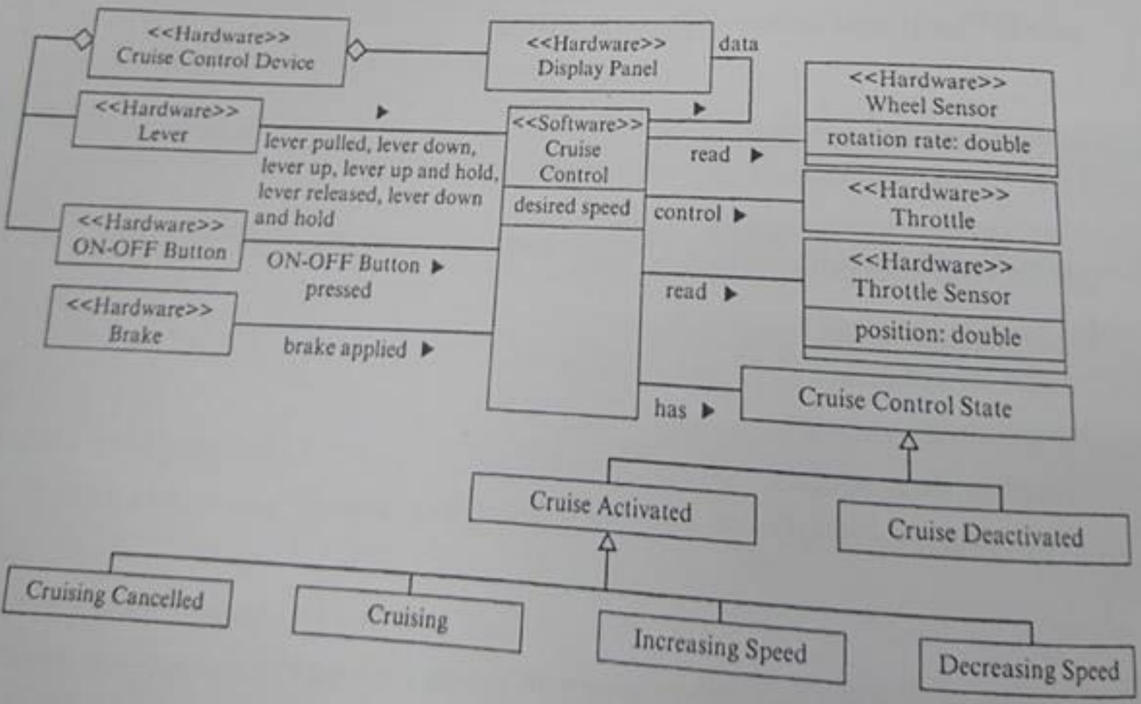
出不去　　到不了　　guard condition

	State & Substate \ Event	Lever down	Lever up and hold	Lever down and hold	Lever released	Lever pulled	Brake applied	Lever up	ON-OFF button pressed
	Cruise deactivated (init)	NA	NA	NA	NA	NA	NA	NA	Cruise activated
Cruise Activated	Cruising canceled (init)	Cruising/ set desired speed	Increasing speed	Decreasing speed	NA	NA	NA	Cruising	Cruise deactivated
	Cruising	Cruising/ set desired speed	Increasing speed	Decreasing speed	NA	Cruising canceled	Cruising canceled	NA	
	Increasing speed	NA	NA	NA	Cruising/set desired speed	NA	NA	NA	
	Decreasing speed	NA	NA	NA	Cruising/set desired speed	NA	NA	NA	

4. We have given the Cruise control domain model and Cruise control state diagram as shown below. Suppose the Cruising state needs to add two lower-level states: Coasting, and Accelerating. If the wheel sensor indicates that the set speed is reached, the cruise control enters the Coasting state. If the wheel sensor shows that the speed is lower than the set speed, the cruise control instructs the gas throttle to accelerate and enters the Accelerating state. Please do the following: 1)

draw the state diagram of the Cruising state. [10%] 2) draw the class diagram of the revised Cruise control state pattern. [10%]





5. Your company web servers are around the world including an Asia server, a Euro server, and an US server. When something goes wrong, you use a crisis center to connect the server, run diagnostics, reboot the server, shutdown the server, and disconnect the server. Now we are going to implement the Command pattern to decouple the object that invokes the operation from the one that knows how to perform it. We will use Asia server and the shutdown command as an example. Please fill in the blank with appropriate Java code. Finally print out the result. 20%

```
public interface Command
{
  Receiver receipver;
①
}

public class ShutDownCommand implements Command
{
Receiver receiver; ②

public ShutDownCommand(Recever ~~receive~~ r) ③
{
 receiver = r; ④
}

public void execute()
{
  receiver.connect();
  receiver.shutdown();
  receiver.disconnect();
  System.out.println();
}

public void undo()
{
```

involker → comand→rec

```
~~System.out.printf("undoing.")~~
receiver.connect();
receiver.shutdown();
receiver.disconnect();
System.out.println();
⑤
  }
}
```

```
// The receiver is going to implement the commands
public interface Receiver
{
public void connect();
public void diagnostics();
public void shutdown();
public void rebot();
public void disconnect();
⑥
}
```

```
public class AsiaServer implements Receiver
{
  public AsiaServer() {
  }
  public void connect() {
    System.out.println("You're connected to the Asia server.");
  }
  public void diagnostics() {
    System.out.println("The Asia server diagnostics check out OK.");
  }
  public void shutdown() {
    System.out.println("Shutting down the Asia server.");
  }
```

```
      public void reboot() {
        System.out.println("Rebooting the Asia server.");
      }
      public void disconnect() {
        System.out.println("You're disconnected from the Asia server.");
      }
    }
```

```
package server;
public class Invoker
{
  Command commands[] = new Command[5];
  int position;

 public Invoker()
 {
   position = -1;
 }

public void setCommand(Command c)
{
  if (position < commands.length - 1){
    position++;
    commands[position] = c;
} else {
  for (int loopIndex = 0; loopIndex < commands.length - 2;
    loopIndex++){
    commands[loopIndex] = commands[loopIndex + 1];
 }
commands[commands.length - 1] = c;
}
}

public void run()
{
  commands[position].execute();
}
```

```
    public void undo()
    {
      if (position >= 0){
        commands[position].undo();
      }
      position--;
    }
}
```

```
public class TestCommands
{
  public static void main(String args[])
  {
    TestCommands t = new TestCommands();
  }

public TestCommands()
{
  // Create an Invoker
```

```
Invoker invoker =new Invoker();
```

⑦

```
  // Create an Asia receiver object
```

Receiver asia = ~~Asia reiver~~ new AsiaServer();

⑧

//1. Create shutdown, run diagnostics, and reboot commands with Asia server; 2. run the invoker for **shutdown** and **reboot** commands, and 3. undo the previous two commands.

```
asia.shutdown();
asia.rundiagnostics();
asia.reboot();
invoker.setCommand(ShutDown);
invoker.setCommand(Reboot);   < invoker.run();
invoker.run();
invoker.undo();
invoker.undo();
```

⑨

//Print out the result