

Dwayne Fraser COP 4331 003
Homework # 4

Problem # 1

Homework 4

5.1

a) How does a Software Engineer use design patterns?

- A Software Engineer uses design patterns to identify a solution that applies the uses and concepts from a pattern.

b) When do you apply the Strategy Pattern

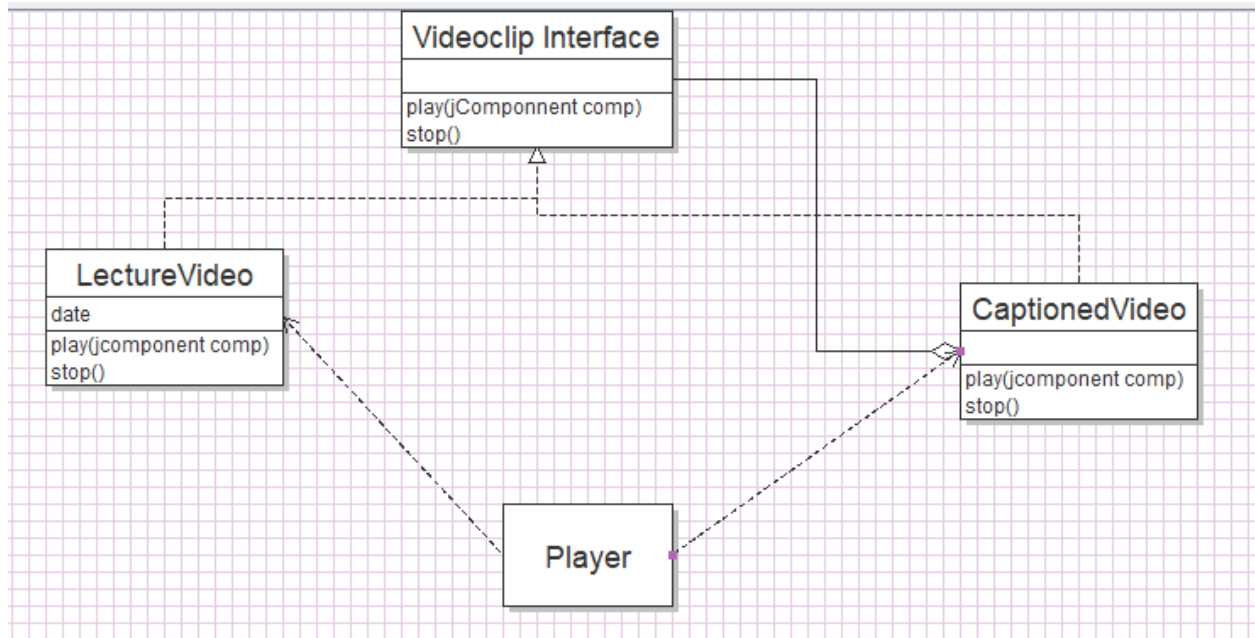
- When selecting an algorithm at runtime

c) The Titled Border class, what design patterns are at work?

- Decorator
- The Titled Border aggregates an object implementing the border interface

Problem # 2

Part A



Part B

Answer: Decorator Pattern

Explanation: CaptionedVideo class implements VideoClip class and aggregates it.

Homework 4

6.1

a) When should Abstract classes be used?

- When using the inheritance concept and using private members.
- Ex: if we want to add new methods in the future to a class

b) When should an interface be used?

- To separate the implementation from the methods

c) What design pattern does class Section implement?

- Composite
- Primitive objects combined

Homework 4

6.2

a) Rectangle2D, Rectangle2D.Float, & Rectangle2D.Double are Template Method because the methods are inherited and implements primitive methods

b) CompoundShape.add is protected because it is intended for subclasses to use them

Problem # 3

```
/*
DWAYNE FRASER
HOMEWORK 4.3
*/

package q3;

import java.awt.geom.*;
import java.awt.*;

public class BarGraph implements Chart, Cloneable {

    public BarGraph(Color color, double min, double max, double val) {
        this.color = color;
        setData(min, max, val);
    }

    public double getData() {
        return value;
    }

    public double getMin() {
        return min;
    }

    public double getMax() {
        return max;
    }

    public void setData(double min, double max, double val) {
        this.min = min;
        this.max = max;
        this.value = val;
    }

    public void setData(double val) {
        this.value = val;
    }

    Color getColor() {
        return color;
    }

    public void draw(Rectangle r, Graphics2D g) {
        double relwidth = (value - min) / (max - min);
        double barwidth = relwidth * r.getWidth();
        Rectangle2D.Double r2 = new Rectangle2D.Double(r.getX(), r.getY(),
            barwidth, r.getHeight());

        g.setColor(color);
        g.fill(r2);
    }

    public BarGraph clone() {
        return new BarGraph(color, min, max, value);
    }
}
```

```

    private Color color;
    private double min;
    private double max;
    private double value;
}

/*
DWAYNE FRASER
HOMEWORK 4.3
*/

package q3;

import java.awt.geom.Rectangle2D;
import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.Rectangle;

public class BarGraph3D extends BarGraph {

    public BarGraph3D(Color col, double min, double max, double val) {
        super(col, min, max, val);
    }

    public void draw(Rectangle r, Graphics2D g) {
        double relwidth = (getData() - getMin()) / (getMax() - getMin());
        double barwidth = relwidth * r.getWidth();
        Rectangle2D.Double r2 = new Rectangle2D.Double(r.getX(), r.getY(),
            barwidth, r.getHeight());

        g.setColor(getColor());
        g.fill3DRect((int)r.getX(), (int)r.getY(), (int)barwidth, (int)r.getHeight(), true);
    }
}

/*
DWAYNE FRASER
HOMEWORK 4.3
*/

```

```

package q3;

import java.awt.Graphics2D;
import java.awt.Rectangle;

public interface Chart {

    void setData(double min, double max, double val);

    void setData(double val);

    double getData();

    public void draw(Rectangle where, Graphics2D g);
}

/*

```

DWAYNE FRASER
HOMEWORK 4.3

*/

package q3;

import java.awt.*;
import javax.swing.*;

```
public class ChartClass extends JPanel {  
  
    public ChartClass(Chart chart) {  
        super(true);  
        this.chart = chart;  
    }  
  
    public void paintComponent(Graphics g) {  
        super.paintComponent(g);  
        Rectangle rv = new Rectangle();  
  
        getBounds(rv);  
  
        Rectangle chartBounds = new Rectangle(0 + HGAP, 0 + VGAP,  
            (int)rv.getWidth()-2*HGAP, (int)rv.getHeight() - 2*VGAP);  
  
        chart.draw(chartBounds, (Graphics2D)g);  
    }  
  
    public void setData(double val) {  
        chart.setData(val);  
    }  
  
    private Chart chart;  
    private final static int HGAP = 8;  
    private final static int VGAP = 8;  
}
```

/*

DWAYNE FRASER
HOMEWORK 4.3

*/

package q3;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import java.awt.BorderLayout;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import javax.swing.BoxLayout;

```
public class ChartPanelSet extends JFrame {  
  
    public ChartPanelSet(Chart[] charts) {  
        int i;  
        int x = charts.length;  
  
        JPanel jpChartPanels = new JPanel();
```

```

jpChartPanels.setLayout(new BorderLayout(jpChartPanels, BorderLayout.Y_AXIS));

ChartClass []panels = new ChartClass[x];
for (i=0; i<x;i++) {
    panels[i] = new ChartClass(charts[i]);
    jpChartPanels.add(panels[i]);
}

JPanel jpNumbers = new JPanel();
jpNumbers.setLayout(new BorderLayout(jpNumbers, BorderLayout.Y_AXIS));

for (i=0; i<x; i++) {
    double val = charts[i].getData();

    JTextField jtfNumber = new JTextField(""+val, 6);

    jtfNumber.addKeyListener(new NumberKeyListener(jtfNumber, panels[i]));
    jpNumbers.add(jtfNumber);
}

setLayout(new BorderLayout());
add(jpNumbers, BorderLayout.EAST);
add(jpChartPanels, BorderLayout.CENTER);

add(new JLabel("Keep numbers in [0,100]"), BorderLayout.NORTH);
}

private class NumberKeyListener implements KeyListener {
    public NumberKeyListener(JTextField tf, ChartClass chartPanel) {
        this.jtfNumber = tf;
        this.chartPanel = chartPanel;
    }

    @Override
    public void keyPressed(KeyEvent e) {
    }

    @Override
    public void keyReleased(KeyEvent e) {
        try {

            double val = Double.parseDouble(jtfNumber.getText());

            chartPanel.setData(val);
            chartPanel.repaint();
        } catch (Exception ex) {

            jtfNumber.selectAll();
        }
    }

    public void keyTyped(KeyEvent e) {
    }

    private JTextField jtfNumber;
    private ChartClass chartPanel;
}
}

```

```

/*
DWAYNE FRASER
HOMEWORK 4.3
*/

package q3;

import javax.swing.*;
import java.awt.Color;
import java.awt.Dimension;

public class Main {

    public static void main(String[] args) {

        Color[] chartColors = new Color[]{Color.green, Color.blue, Color.orange};
        int x = chartColors.length;

        Chart[] charts = new Chart[x];
        int i;

        for (i=0; i<x/2; ++i) {
            charts[i] = new BarGraph3D(chartColors[i], 0, 100, 50);
        }

        for (i=x/2; i<x; ++i) {
            charts[i] = new BarGraph3D(chartColors[i], 0, 100, 50);
        }

        ChartPanelSet y = new ChartPanelSet(charts);

        y.setPreferredSize(new Dimension(600,200));

        y.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        y.pack();
        y.setVisible(true);
    }
}

```

Problem #4

Part A

Mapping

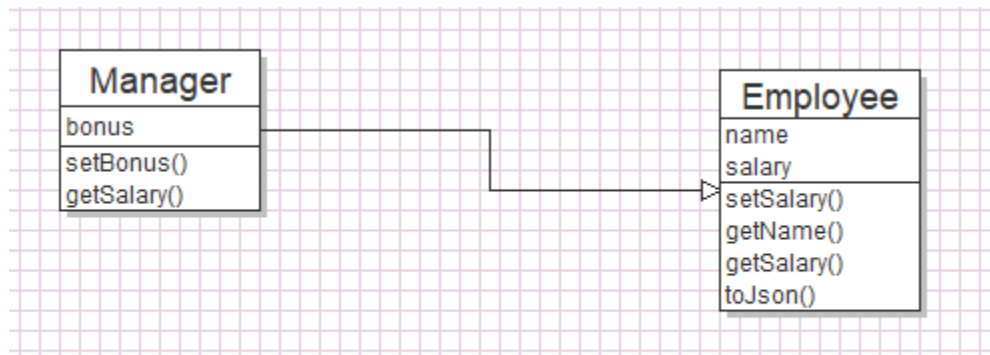
Name in Design Pattern	Actual Name
AbstractClass	
ConcreteClass	Employee, Manager
templateMethod()	toJson()

primitiveOp1(), primitiveOp2(), primitiveOp3()

getName(), getSalary(), setSalary()

Part B

Class Diagram



Part C

Program

```
/*
DWAYNE FRASER
HOMEWORK Ch 6.3
*/

package q4;

public class Employee
{
    public Employee(String aName) { name = aName; }
    public void setSalary(double aSalary) { salary = aSalary; }
    public String getName() { return name; }
    public double getSalary() { return salary; }

    public String toJson() {

        StringBuilder x = new StringBuilder();

        x.append(getName());

        x.append(getSalary());
    }
}
```

```

    return x.toString();

}

private final String name;
private double salary;
}

/*
DWAYNE FRASER
HOMEWORK Ch 6.3
*/

package q4;

public class Manager extends Employee {

    public Manager(String aName) {
        super(aName);
    }

    public void setBonus(double b) {
        bonus = b;
    }

    public double getSalary() {
        return super.getSalary() + bonus;
    }

    private double bonus;

    public static void main(String[] args) {

        Employee sarah = new Employee("Sarah");
        sarah.setSalary(50000);

        Manager sandy = new Manager("Sandy");

        sandy.setSalary(100000);
        sandy.setBonus(1234);

        System.out.println(sarah.toJson());
        // prints {"class":"Employee", "name":"Sarah", "salary": 50000}
        System.out.println(sandy.toJson());
    }
}

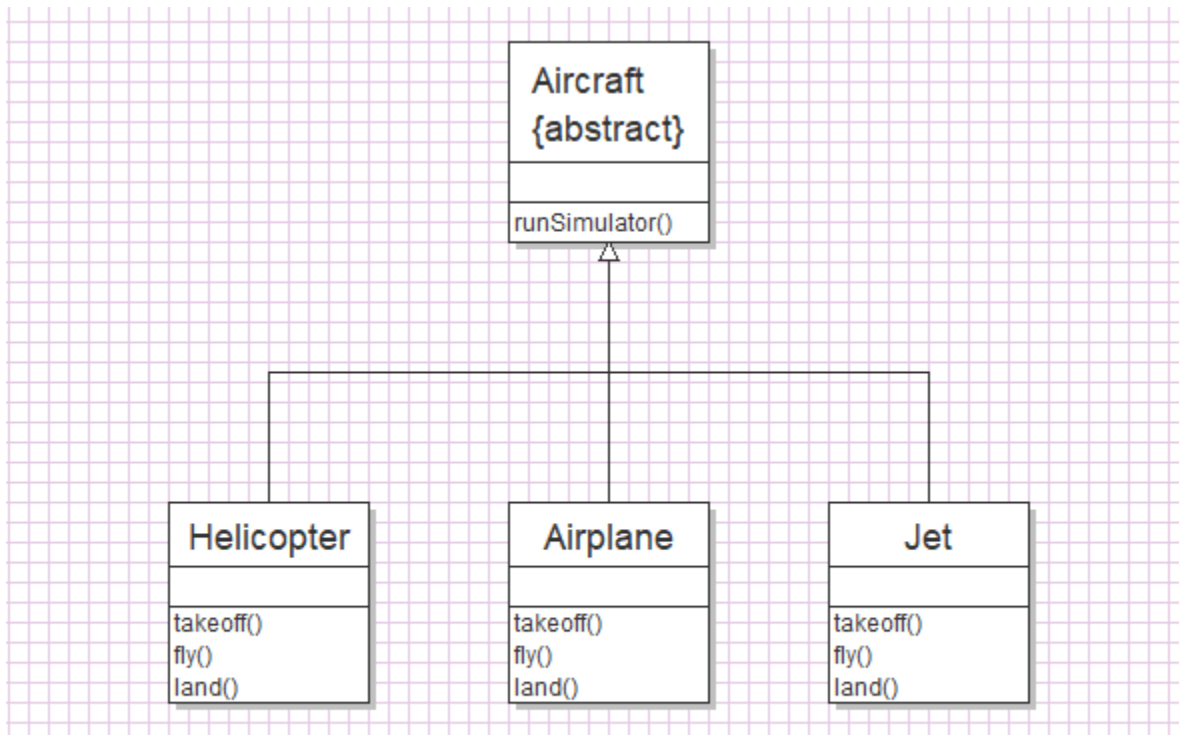
```

```
// prints {"class":"Manager", "name":"Sandy", "salary": 101234, "bonus":1234}  
}  
}
```

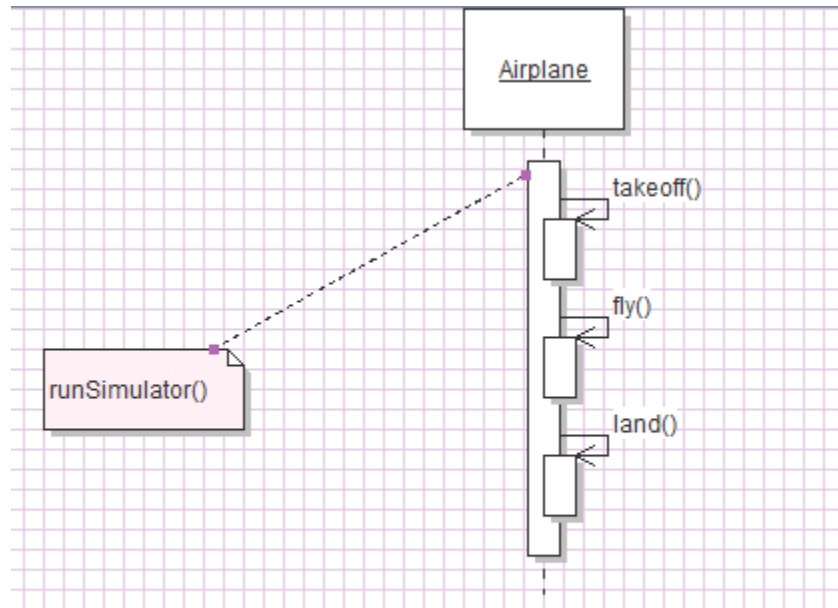
Problem #5

Part A

Class Diagram



Sequence Diagram



Part B

Program

```
/*
DWAYNE FRASER
HOMEWORK Ch 6.4
*/

package q5;

public abstract class Aircraft {

    public void runSimulator() {
        takeOff();
        fly();
        land();
    }

    public abstract void takeOff();
    public abstract void fly();
    public abstract void land();

}

/*
DWAYNE FRASER
HOMEWORK Ch 6.4
*/

package q5;

public class Airplane extends Aircraft {
```

```

    public void takeOff() {
        System.out.println("Success");
    }

    public void fly() {
        System.out.println("Success");
    }

    public void land() {
        System.out.println("Success");
    }
}

/*
DWAYNE FRASER
HOMEWORK Ch 6.4
*/

```

```

package q5;

public class Jet extends Aircraft {

    public void takeOff() {
        System.out.println("Success");
    }

    public void fly() {
        System.out.println("Success");
    }

    public void land() {
        System.out.println("Success");
    }
}

/*
DWAYNE FRASER
HOMEWORK Ch 6.4
*/

```

```

package q5;

public class Helicopter extends Aircraft {

    public void takeOff() {
        System.out.println("Success");
    }

    public void fly() {
        System.out.println("Success");
    }

    public void land() {
        System.out.println("Success");
    }
}

```

```
/*  
DWAYNE FRASER  
HOMEWORK Ch 6.4  
*/  
  
package q5;  
  
public class runSimulator {  
  
    public static void main(String[] args) {  
  
        Aircraft x = new Airplane();  
        x.runSimulator();  
  
        Aircraft y = new Jet();  
        y.runSimulator();  
  
        Aircraft z = new Helicopter();  
        z.runSimulator();  
  
    }  
}
```