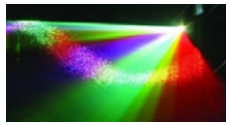
Supplement to
The Design and Implementation of Multimedia Software

The Strategy Pattern

Prof. David Bernstein

James Madison University

users.cs.jmu.edu/bernstdh



Motivation

- The Setting:

  In most situations there are many ways to achieve the same result

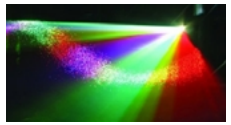
  In many of these situations, the software designer chooses one (hopefully the best) and implements it

  In some situations, one wants to have the flexibility to use more than one

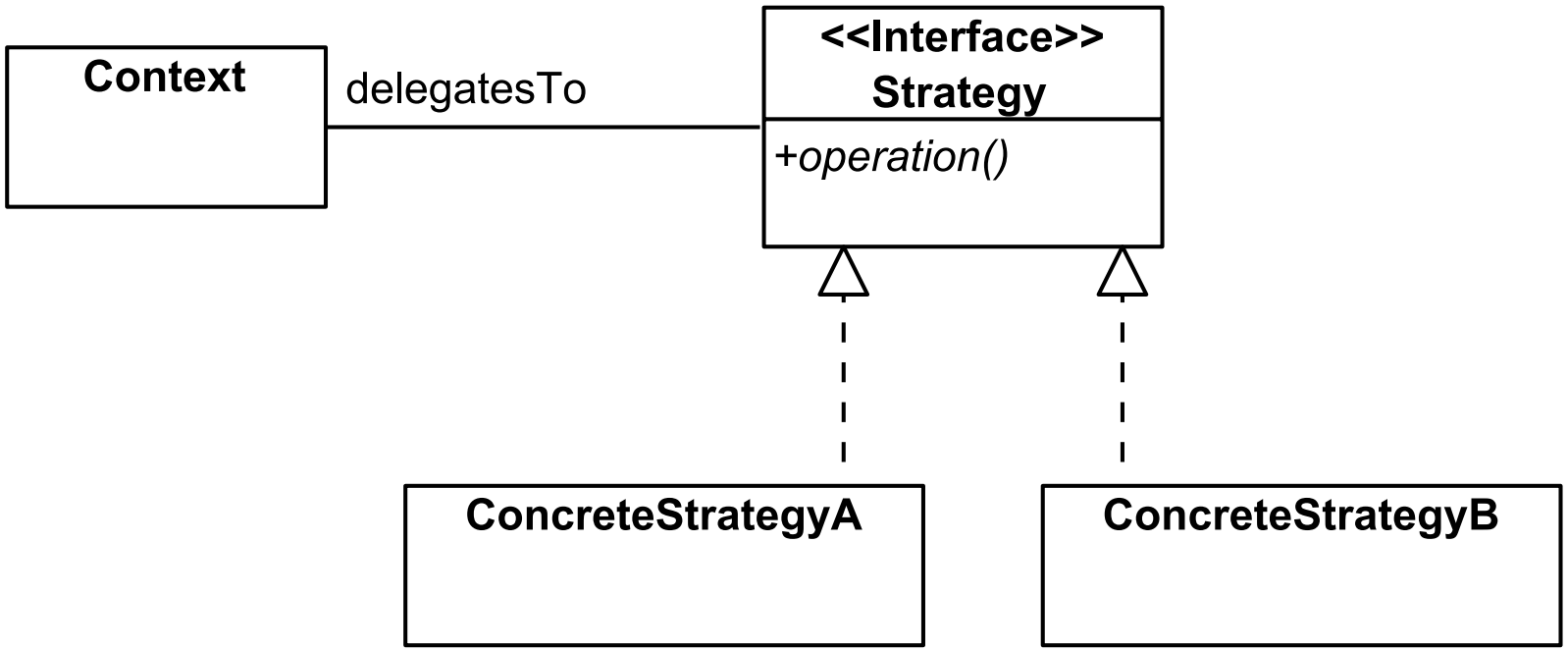
- Examples:

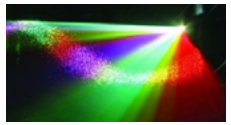
  Multidimensional search algorithms can use many one-dimensional search algorithms

  A document formatter can use many paragraph formatters

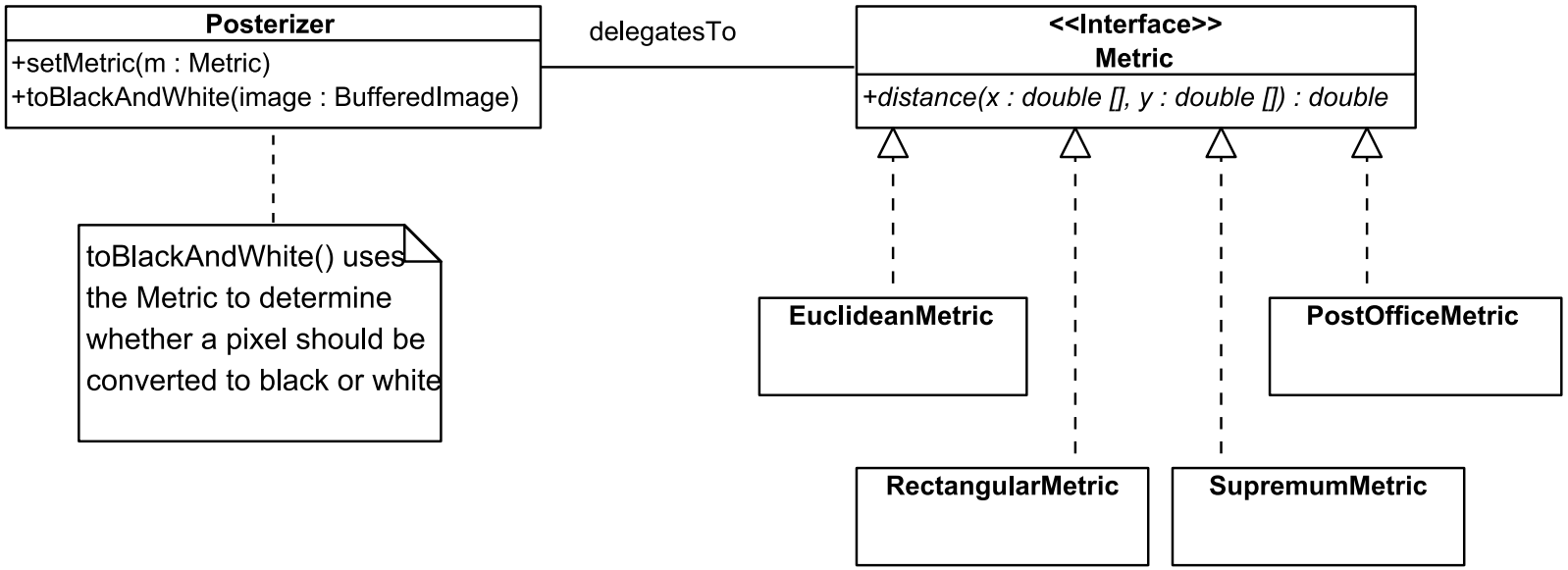


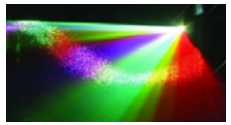
The Strategy Pattern





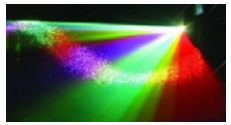
An Example


Posterizer
+setMetric(m : Metric)
+toBlackAndWhite(image : BufferedImage)
delegatesTo
<<Interface>>
Metric
+distance(x : double [], y : double []) : double
toBlackAndWhite() uses the Metric to determine whether a pixel should be converted to black or white
EuclideanMetric
PostOfficeMetric
RectangularMetric
SupremumMetric



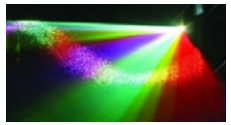
An Example (cont.)

```
package math;

public interface Metric
{
    public abstract double distance(double[] x, double[] y);
}
```



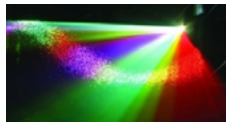
An Example (cont.)

```
package math;

public class      EuclideanMetric
       implements Metric
{
    public double distance(double[] x, double[] y)
    {
       double  result;
       int     n;

       result = 0.0;
       n      = Math.min(x.length, y.length);

       for (int i=0; i<n; i++)
       {
          result += Math.pow(x[i]-y[i], 2.0);
       }

       return Math.sqrt(result);
    }
}
```



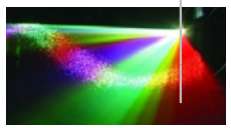
An Example (cont.)

```
package math;

public class      SupremumMetric
       implements Metric
{
    public double distance(double[] x, double[] y)
    {
       double  result, term;
       int     n;

       result = Math.abs(x[0]-y[0]);
       n      = Math.min(x.length, y.length);

       for (int i=1; i<n; i++)
       {
          term = Math.abs(x[i]-y[i]);

          if (term > result) result = term;
       }

       return result;
    }
}
```



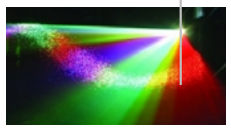
An Example – The Driver

```
import java.awt.*;
import java.awt.image.*;

import math.*;


public class Posterizer
{
    private double[]    x, y;
    private Metric      metric;


    private static final int[] BLACK = {  0,  0,  0};
    private static final int[] WHITE = {255,255,255};

    public Posterizer()
    {
       x = new double[3];
       y = new double[3];
    }


    private double distance(int[] a, int[] b)
    {
       double     result;

       for (int i=0; i<3; i++)
       {
```



An Example – The Driver (cont.)

```
            x[i] = a[i];
            y[i] = b[i];
        }

        result = Double.POSITIVE_INFINITY;
        if (metric != null) result = metric.distance(x, y);

        return result;
    }


    public void setMetric(Metric metric)
    {
        this.metric = metric;
    }


    public void toBlackAndWhite(BufferedImage image)
    {
        ColorModel  colorModel;
        double      blackDistance, whiteDistance;
        int         height, packedPixel, packedBlack, packedWhite, width;
        int[]       pixel;

        pixel       = new int[3];

        height      = image.getHeight();
        width       = image.getWidth();
```



An Example – The Driver (cont.)

```
        colorModel  = image.getColorModel();

        packedBlack = colorModel.getDataElement(BLACK,0);
        packedWhite = colorModel.getDataElement(WHITE,0);

        for (int x=0; x<width; x++)
        {
           for (int y=0; y<height; y++)
           {
              packedPixel = image.getRGB(x, y);
              colorModel.getComponents(packedPixel, pixel, 0);

              blackDistance = distance(pixel, BLACK);
              whiteDistance = distance(pixel, WHITE);

              if (blackDistance < whiteDistance)
                 image.setRGB(x, y, 0x00000000);
              else
                 image.setRGB(x, y, 0xFFFFFFFF);
           }
        }
     }

}
```