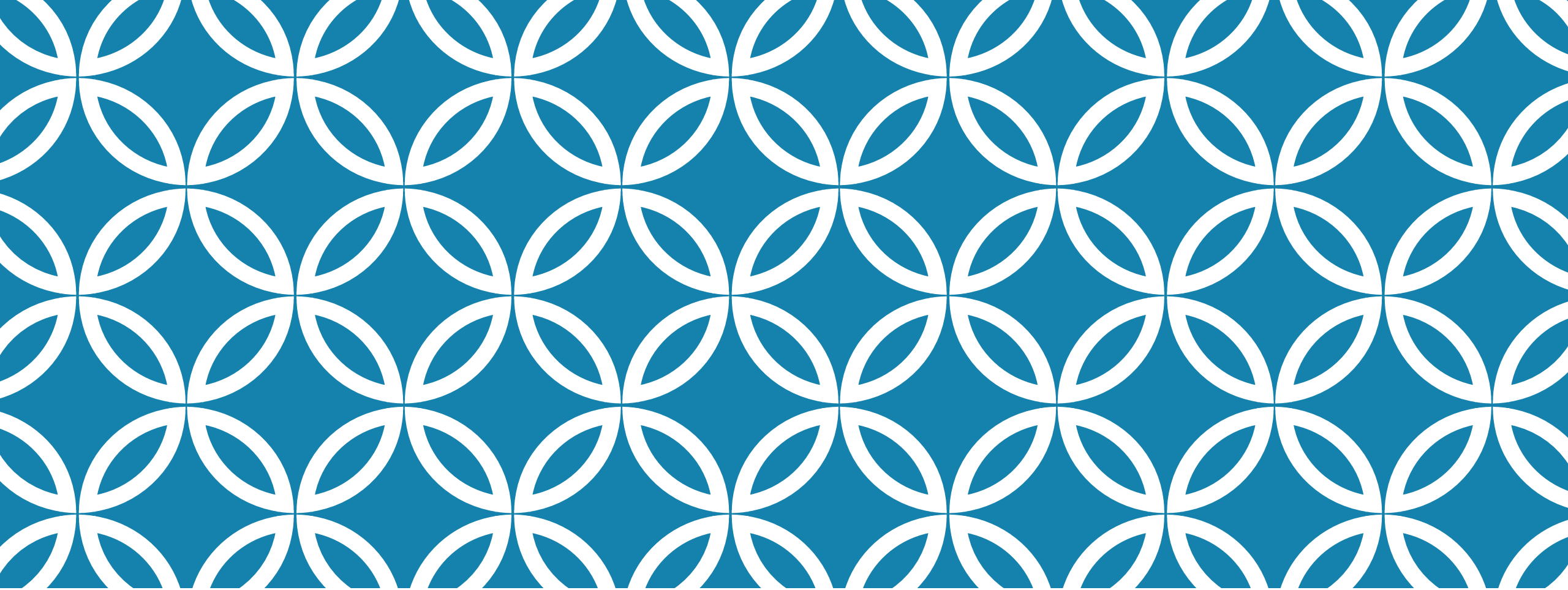

RECITATION 4

Casting and graphics (with ObjectEditor)

CASTING

Changing the type of one variable into another type

Cast a variable by adding the new type in parentheses before the variable

Example:

```
int ex = 5;
System.out.println(ex/2);  //PRINTS 2
System.out.println((double)ex / 2);  //PRINTS 2.5
System.out.println((double)(ex/2)); //PRINTS 2.0
```

Make sure that you are using parentheses carefully!
Casting only affects the token directly after it

DRAWING WITH OBJECT EDITOR

Create a Bean with the editable properties: Height, Width, X, and Y

Tag the bean with @StructurePattern(StructurePatternNames.****_PATTERN)

- Where **** is the shape that you want: line, rectangle, or oval

Use the OEFrame.refresh() method in order to update the position of the shapes (after changing the properties)

CODING "ASSIGNMENT"

Create a new project or use an existing project and either:

Create a new package called recitation4, we might change this later, copy and paste the code from the end of this PowerPoint into their respective classes (Driver, IShapePair, IResettable, IShape, AShape, ALine, and ShapePair)

Or

Save the .zip on the recitation page, right click on the src folder for the project you want to use in eclipse, select import, General → Archive File, From Archive Browse (find the recitation4.zip), Into Folder should say: ****/src/, finish.

Either way you chose, make sure you add oeall22 as an external JAR to your project

We will fiddle around with the commented lines and the constants.

```java
package recitation4;

import util.annotations.StructurePattern;
import util.annotations.StructurePatternNames;

@StructurePattern(StructurePatternNames.LINE_PATTERN)
public class ALine implements IShape, IResettable {
int x, y, width, height, initX, initY, initHeight, initWidth;

public ALine(int initX, int initY, int initWidth, int initHeight) {
this.initX = initX;
        this.initY = initY;
        this.initWidth = initWidth;
        this.initHeight = initHeight;
        setX(initX);
        setY(initY);
        setHeight(initHeight);
        setWidth(initWidth);
}
public int getX() {
return x;
}
public void setX(int newX) {
x = newX;
}
public int getY() {
return y;
}
public void setY(int newY) {
y = newY;
}
public int getWidth() {
return width;
}
public void setWidth(int newVal) {
width = newVal;
}
public int getHeight() {
return height;
}
public void setHeight(int newHeight) {
height = newHeight;
}
@Override
public void reset() {
setHeight(initHeight);
setWidth(initWidth);
setX(initX);
setY(initY);
}
}
```

```java
package recitation4;

public interface IResettable {
public void reset();
}
```

```java
package recitation4;

public interface IShape {
    public int getX();
    public void setX(int newVal);
    public int getY();
    public void setY(int newVal);
    public int getWidth();
    public void setWidth(int newVal);
    public int getHeight() ;
    public void setHeight(int newVal);

}
```

```java
package recitation4;

public interface IShapePair {
public void setFirst(IShape newFirst);
public IShape getFirst();
public void setSecond(IShape newSecond);
public IShape getSecond();
}
```

```java
package recitation4;

import util.annotations.ComponentWidth;
import util.annotations.StructurePattern;
import util.annotations.StructurePatternNames;

import java.awt.Color;

//@StructurePattern(StructurePatternNames.LINE_PATTERN)
//@StructurePattern(StructurePatternNames.RECTANGLE_PATTERN)
//@StructurePattern(StructurePatternNames.OVAL_PATTERN)
//@StructurePattern(StructurePatternNames.BEAN_PATTERN)

/******^^DO NOT UNCOMMENT MORE THAN ONE OF THE ABOVE AT A TIME ^^*****/

public class AShape implements IShape, IResettable{
    int x, y, width, height, initWidth, initHeight, initX, initY;

    public AShape (int initX, int initY, int initWidth, int initHeight) {
        this.initX = initX;
        this.initY = initY;
        this.initWidth = initWidth;
        this.initHeight = initHeight;
        setX(initX);
        setY(initY);
        setHeight(initHeight);
        setWidth(initWidth);
    }
//    public Color getColor(){return Color.blue;}
//    @ComponentWidth(50)    // only affects main panel (text fields)
    public int getX() {return x;}
    public void setX(int newX) {x = newX;}
    public int getY() {return y;}
    public void setY(int newY) {y = newY;}
    public int getWidth() {return width;}
    public void setWidth(int newVal) {width = newVal;}
//    @ComponentWidth(250)    // only affects main panel (text fields)
    public int getHeight() {return height;}
    public void setHeight(int newHeight) {height = newHeight;}

    public void reset() {
    setX(initX);
    setY(initY);
    setHeight(initHeight);
    setWidth(initWidth);
    }

}
```

```java
package recitation4;

public class ShapePair implements IShapePair {
IShape first;
IShape second;

public ShapePair(IShape initFirst, IShape initSecond) {
setFirst(initFirst);
setSecond(initSecond);
}

@Override
public void setFirst(IShape newFirst) {
first = newFirst;
}

@Override
public IShape getFirst() {
return first;
}

@Override
public void setSecond(IShape newSecond) {
second = newSecond;
}

@Override
public IShape getSecond() {
return second;
}

}
```

```java
package recitation4;
import bus.uigen.OEFrame;
import bus.uigen.ObjectEditor;

public class Driver {
private static final int NUM_MOVES = 100;
private static final int SLEEP_TIME = 10;
private static final int INIT_LINE_X = 10;
private static final int INIT_LINE_Y = 50;
private static final int INIT_LINE_WIDTH = 30;
private static final int INIT_LINE_HEIGHT = 30;
private static final int INIT_SHAPE_X = 50;
private static final int INIT_SHAPE_Y = 50;
private static final int INIT_SHAPE_HEIGHT = 30;
private static final int INIT_SHAPE_WIDTH = 30;
private static final int EDITOR_X = 800;
private static final int EDITOR_Y = 300;

public static void main(String[] args) {
IShapePair pair = new ShapePair(new ALine(INIT_LINE_X, INIT_LINE_Y, INIT_LINE_WIDTH, INIT_LINE_HEIGHT),
new AShape(INIT_SHAPE_X, INIT_SHAPE_Y, INIT_SHAPE_WIDTH, INIT_SHAPE_HEIGHT));
OEFrame editor = ObjectEditor.edit(pair);
//editor.showTreePanel();   /** Try uncommenting this **/
//editor.hideMainPanel();   /** Try uncommenting this (when using Bean Pattern in AShape) **/
editor.setSize(EDITOR_X, EDITOR_Y);
/*demo, motion of object*/
for(int i = 0;i < NUM_MOVES; i++){
sleep(SLEEP_TIME);/* the unit is millisecond*/
moveShapes(pair,1);
editor.refresh();
}
resetPair(pair.getFirst(), editor);
resetPair(pair.getSecond(), editor);
/*demo ends*/

}

public static void resetPair(IShape shape, OEFrame editor) {
((IResettable)shape).reset();
editor.refresh();
}
/*move an object along the x axis*/
public static void moveShape(IShape lineInstance, int xDist){
lineInstance.setX(xDist + lineInstance.getX());
}
/* move an array of objects along the x axis
 *  by calling the moveShape method individually
 */
public static void moveShapes(IShapePair pair, int xDist){
moveShape(pair.getFirst(), xDist);
moveShape(pair.getSecond(), xDist);
}

/*enable program can be hung up for a while*/
    public static void sleep(long interval) {
        try {
            Thread.sleep(interval);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```