Lab 2: **SletchPad: An Application of Polymorphism**

{See Sanders & Van Dam: Chapter 5 [5.4]: pp 166 -173]

**PreLab: DownLoad wheels.zip and add it to your l ibrary in BlueJ:**

### Part I: Download wheels.zip to a convenient folder or to the desktop

## Part II: Start up BlueJ Open Preferences under the tools menu

### Choose Libraries tab

### Click Add button next to user libraries

### Find your  wheels.zip file Click Open

### At this point, the wheels.zip file should be shown under user libraries

### Hit the OK button

### Quit BlueJ and then reopen BlueJ

## Lab 2: Login and start BlueJ.

## Create a project called Sketch.

## Create the following classes:

### SketchApp

### DrawButton

### UpButton

### DownButton

### LeftButton

### RightButton

### Cursor

## 4) Modify each of the classes so that they contain the code shown below:

|  |
| --- |
| **SketchApp Class**  import wheels.users.\*;  public class SketchApp extends Frame {    private Cursor \_cursor;    private DrawButton \_upButton, \_downButton;    private DrawButton \_leftButton, \_rightButton;        public SketchApp() {       \_cursor = new Cursor();       \_upButton = new UpButton(350, 400, \_cursor);       \_rightButton = new RightButton(370, 420, \_cursor);       \_leftButton = new LeftButton(330, 420, \_cursor);       \_downButton = new DownButton(350, 440, \_cursor);    }         public static void main() {       SketchApp myPad = new SketchApp();    } }  **DrawButton Class**  import wheels.users.\*;  public class DrawButton extends Ellipse {    private Cursor \_cursor;        public DrawButton(int x, int y, Cursor cursor) {       super(x, y);       this.setSize(20, 20);       \_cursor = cursor; //store reference to peer cursor    }        public void mousePressed(java.awt.event.MouseEvent e) {       this.setFillColor(java.awt.Color.BLUE);    }        public void mouseReleased(java.awt.event.MouseEvent e) {       java.awt.Point lastPoint = \_cursor.getLocation();       java.awt.Point nextPoint = computeNextPoint(lastPoint);       Line line = new Line(lastPoint, nextPoint);       line.setColor(java.awt.Color.BLACK);       line.setThickness(2);       \_cursor.setLocation(nextPoint);       this.setFillColor(java.awt.Color.RED);    }        public java.awt.Point computeNextPoint(java.awt.Point lastPoint) {       return new java.awt.Point(0, 0); // default: move cursor to (0,0)    } }  **UpButton Class**  import wheels.users.\*;  public class UpButton extends DrawButton {    public UpButton (int x, int y, Cursor cursor) {       super(x, y, cursor);    }        public java.awt.Point computeNextPoint(java.awt.Point lastPoint) {       return new java.awt.Point(lastPoint.x, lastPoint.y-5);    } }  **DownButton Class**  import wheels.users.\*;  public class DownButton extends DrawButton {    public DownButton (int x, int y, Cursor cursor) {       super(x, y, cursor);    }        public java.awt.Point computeNextPoint(java.awt.Point lastPoint) {       return new java.awt.Point(lastPoint.x, lastPoint.y+5);    } }  **LeftButton Class**  import wheels.users.\*;  public class LeftButton extends DrawButton {    public LeftButton (int x, int y, Cursor cursor) {       super(x, y, cursor);    }        public java.awt.Point computeNextPoint(java.awt.Point lastPoint) {       return new java.awt.Point(lastPoint.x-5, lastPoint.y);    } }  **RightButton Class**  import wheels.users.\*;  public class RightButton extends DrawButton {    public RightButton (int x, int y, Cursor cursor) {       super(x, y, cursor);    }        public java.awt.Point computeNextPoint(java.awt.Point lastPoint) {       return new java.awt.Point(lastPoint.x+5, lastPoint.y);    } }  **Cursor Class**  import wheels.users.\*;  public class Cursor {    private java.awt.Point \_location;        public Cursor () {       \_location = new java.awt.Point(350, 250);    }        public void setLocation(java.awt.Point point) {       \_location = point;    }        public java.awt.Point getLocation() {       return \_location;    } } |

## 5) Take a Screen shot of a drawing that you did and paste it here:

|  |
| --- |
|  |

## 6) Modify the program so that it displays eight buttons (instead of four) arranged like this:

|  |  |  |  |
| --- | --- | --- | --- |
| O | O | O | **Note: You will need to create 4 additional classes.** The **corner buttons** |
| O |  | O | should each make the line go diagonally (i.e. 5 pixels in the x direction |
| O | O | O | and 5 **[**pixels in the y direction. ). The rest of the buttons should continue to work as they do now |

### 7. Compile, debug, and run your program.

### .

## 8) Take a Screen shot of a drawing that you did and paste it here:

|  |
| --- |
|  |

Upload this document along with a zipped file of your completed project to Blackboard. This project is due no later than Thursday, February 6th by midnight.