 

**Advanced Placement Computer Science**

[**Shenendehowa HS**](http://www.shenet.org/shen-high-school/)[**mr Hanley**](http://hanley.co.nr)

**Unit 4: Object Oriented ProgrammingFILLED\_OUT**

**Lesson: Advanced Features Ver 3.0**

***Last Updated:*** *11/13/2017*

Lesson: Parameter Passing Mechanisms

*Last Updated: 100/11001/1100*

In addition to writing classes using private variables and public accessors and mutators, here are 3 common features that are built into classes.

1. Exception Handling: A way of responding to error conditions

In your class, add the statement throws with a type of Exception

In your client, add a try catch block
java will search for a try catch block in previously called methods
2. static or class variables:
variables SHARED by all instances of the class
3. public String toString()
represents object as a single string
Useful for debugging!!!!

/\*\*

 \* ---------------------------------------------------------------------------

 \* S-h-e-n-e-n-d-e-h-o-w-a--H-i-g-h--S-c-h-o-o-l--T-e-c-h-n-o-l-o-g-y--D-e-p-t

 \* ---------------------------------------------------------------------------

 \* FILE: RectangleV2.java

 \* DATE: 11/23/2013

 \* PURPOSE: Demonstrate the Comparable interface, toString, static

 \* variables and exception handling

 \* @author mr Hanley

 \* @version 1.0

 \* ----------------------------------------------------------------------------

 \* h-a-n-l-e-y.c-o-.-n-r------t-e-a-m-2-0-.-c-o-m------------------------------

 \*/

import java.awt.Color;

import java.awt.Graphics;

public class RectangleV2 implements Comparable {

 //++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

 //++++++++++++ S T A T I C V A R I A B L E S ++++++++++++++++

 //++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

 public static int rectCount = 0;//Yo yo yo, this is SHARED BY ALL //RECTANGLESV2

 //----------------------------------------------------------------

 //---------------- I N S T A N C E V A R I A B L E S ------------

 //----------------------------------------------------------------

 private int x, y, width, height, rotation; //rotation is 0 deg

 private Color outCol, fillColl;

 private boolean filled;

 private FillType ft;

 //////////////////////////////////////////////////////////////////

 //////// C O N S T R U C T O R S ///////////////////////

 //////////////////////////////////////////////////////////////////

 public RectangleV2() {

 x = y = 0;

 width = 50;

 height = 25;

 outCol = Color.BLACK;

 fillColl = Color.CYAN;

 filled = true;

 ft = FillType.solid;

 rectCount++;
}

 public RectangleV2(Color outColIn, Color fillCollIn) {

 x = y = 0;

 width = 50;

 height = 25;

 outCol = outColIn;

 fillColl = fillCollIn;

 filled = true;

 ft = FillType.solid;

 rectCount++;

 }

 public RectangleV2(int xIn, int yIn, int w, int h) {

 x = xIn;

 y = yIn;

 width = w;

 height = h;

 outCol = Color.black;

 fillColl = Color.MAGENTA;

 filled = true;

 ft = FillType.solid;

 rectCount++;

 }

 //AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

 //AAAAAAAAAAAAAAA A C C E S S O R S AAAAAAAAAAAAAAAAAAAAAAAAAAAAA

 //AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

 //Compareto is based on position, row major

 public int compareTo(Object other) {

 Rectangle otherR = (Rectangle) other; //typecast

 if (y != otherR.getY()) {

 return y - otherR.getY();

 }

 //Else go to the column

 return x - otherR.getX();

 }

 public int getX() {

 return x;

 }

 public int getY() {

 return y;

 }

 public int getW() {

 return width;

 }

 public int getH() {

 return height;

 }

 public Color getOutCol() {

 return outCol;

 }

 public Color getFillCol() {

 return fillColl;

 }

 public int getArea() {

 return width \* height;

 }

 public String toString() {

 return "[" + x + "," + y + " Width = " + width + " Height = " + height + " Total number of rect= "+rectCount +"]";//Brian

 }

 public void draw(Graphics g) {

 //Draw the first rectangle

 //System.out.println("Here");

 g.setColor(getOutCol());

 //System.out.println("x = "+getX()+"width"+width);

 g.drawRect(getX(), getY(), width, height);

 g.setColor(getFillCol());

 g.fillRect(getX() + 1, getY() + 1, width - 1, height - 1);

 }

 //MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM

 //MMMMMMMMMMMMMMM M U T A T O R S MMMMMMMMMM

 //MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM

 public void setX(int inX) throws IllegalArgumentException {

 if (inX < 0) {

 throw new IllegalArgumentException("Bad x value of " + inX);

 }

 x = inX;

 }

 public void setY(int inY) {

 y = inY;

 }

 public void setWidth(int w) {

 width = w;

 }

 public void setHeight(int h) {

 height = h;

 }

}