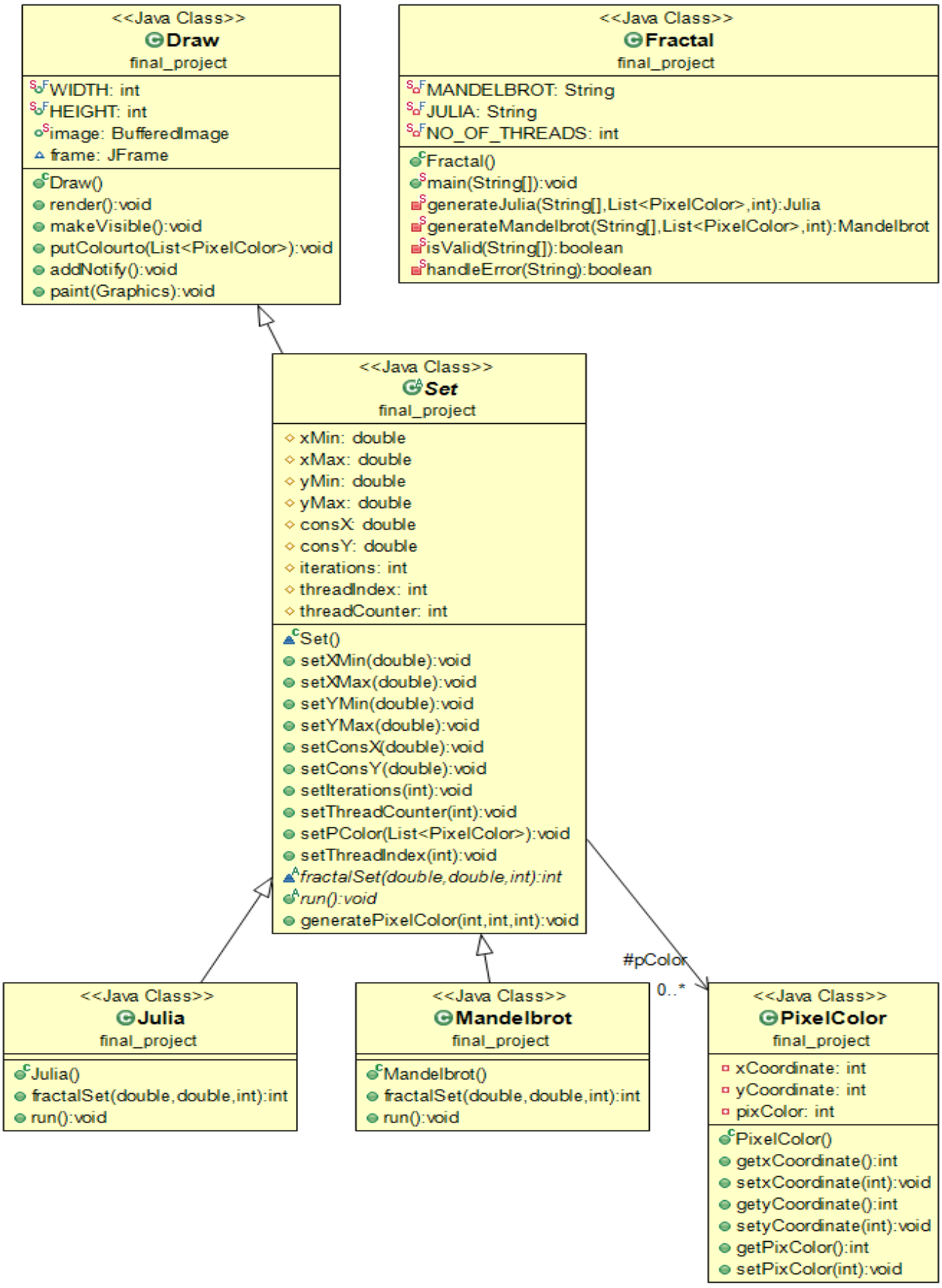
<<Java Class>>
Draw
final_project
WIDTH: int
HEIGHT: int
image: BufferedImage
frame: JFrame
Draw()
render():void
makeVisible():void
putColourto(List<PixelColor>):void
addNotify():void
paint(Graphics):void
<<Java Class>>
Fractal
final_project
MANDELBROT: String
JULIA: String
NO_OF_THREADS: int
Fractal()
main(String[]):void
generateJulia(String[],List<PixelColor>,int):Julia
generateMandelbrot(String[],List<PixelColor>,int):Mandelbrot
isValid(String[]):boolean
handleError(String):boolean
<<Java Class>>
Set
final_project
xMin: double
xMax: double
yMin: double
yMax: double
consX: double
consY: double
iterations: int
threadIndex: int
threadCounter: int
Set()
setXMin(double):void
setXMax(double):void
setYMin(double):void
setYMax(double):void
setConsX(double):void
setConsY(double):void
setIterations(int):void
setThreadCounter(int):void
setPColor(List<PixelColor>):void
setThreadIndex(int):void
fractalSet(double,double,int):int
run():void
generatePixelColor(int,int,int):void
#pColor
0..*
<<Java Class>>
Julia
final_project
Julia()
fractalSet(double,double,int):int
run():void
<<Java Class>>
Mandelbrot
final_project
Mandelbrot()
fractalSet(double,double,int):int
run():void
<<Java Class>>
PixelColor
final_project
xCoordinate: int
yCoordinate: int
pixColor: int
PixelColor()
getxCoordinate():int
setxCoordinate(int):void
getyCoordinate():int
setyCoordinate(int):void
getPixColor():int
setPixColor(int):void