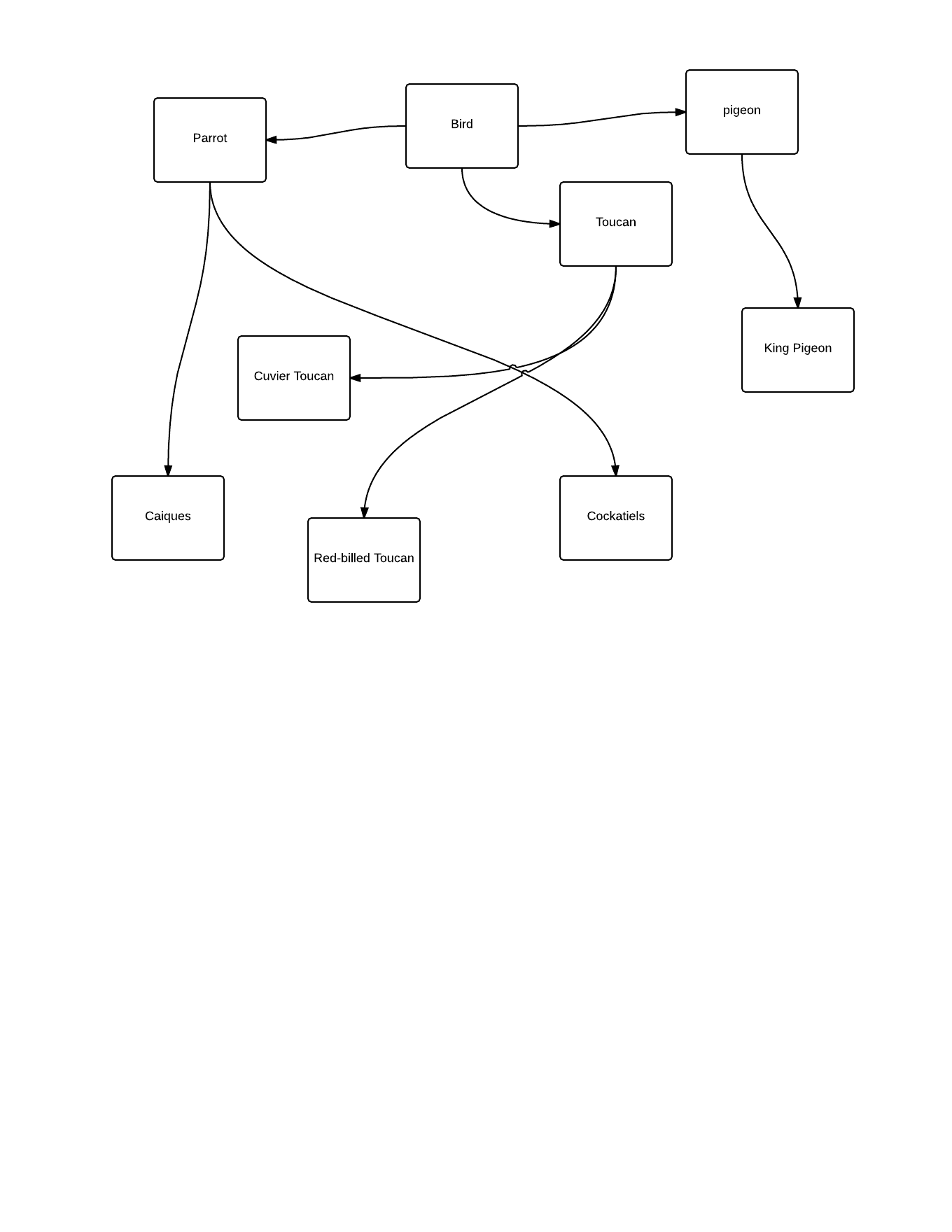
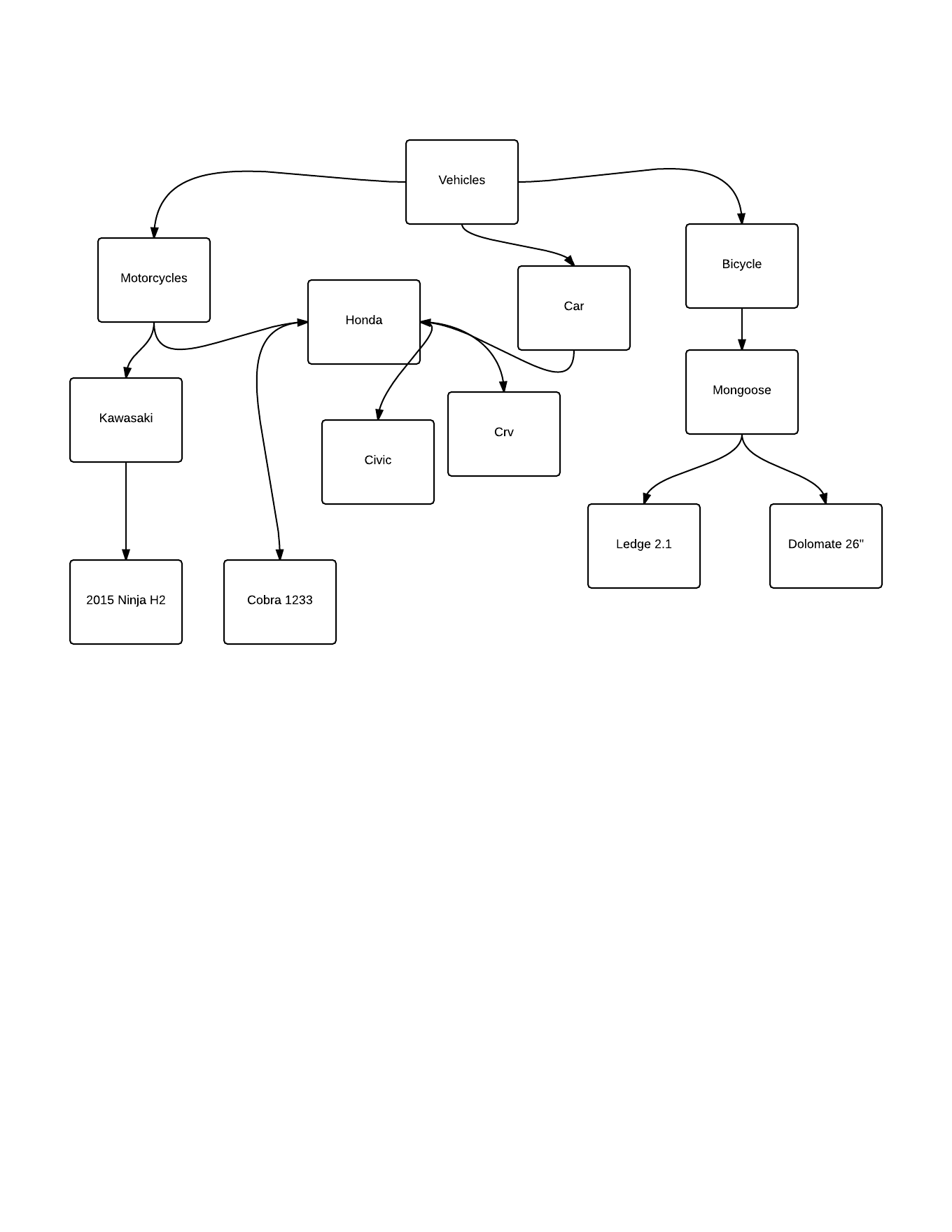
## Project 11-1



## Project 11.2



## Project 11.3

//shape.java

**public** **double** perimeter();

//Abstract shape.java

**abstract** **public** **double** perimeter();

///Circle . java

**public** **double** perimeter(){

**return** 2 \*Math.***PI*** \* radius ;

}

## Project 11.4

//Abstract shape.java

**public** **void** move (**double** xLoc, **double** yLoc){

xPos = xLoc;

yPos = yLoc;

}

// project 11-4

**public** **class** Triangle **extends** AbstractShape {

**private** **double** x1,x2,y1,y2;

**public** Triangle() {

**super**();

x1=2;

y1 =2;

y2=0;

}

**public** Triangle(**double** xLoc, **double** yLoc, **double** x, **double** y, **double** a, **double** {

**super**(xLoc, yLoc);

x1 =x;

y1 =y;

x2 =a;

y2=b;

}

**public** **double** area() {

**return** 0.5 \* Math.*abs*(getXPos()\*y1 -x1 \* getYPos() + x1\* y2-x2 \* y1 +x2 \*getYPos()-getXPos()\* y2);

}

**public** **double** distance(){

**return** Math.*sqrt*((getXPos() - x1)\* (getXPos() - x1) + (getYPos() -y1) \* (getYPos() -y1));

}

**public** **double** perimeter() {

**double** side1 = Math.*sqrt*(Math.*pow*((x2-x1),2)\*Math.*pow*((y2-y1),2));

**double** side2 = Math.*sqrt*(Math.*pow*((getXPos()-x2),2)\*Math.*pow*((getYPos()-y2),2));

**double** side3 = Math.*sqrt*(Math.*pow*((getXPos()-x1),2)\*Math.*pow*((getYPos()-y1),2));

**return** side1 +side2 + side3;

}

**public** **void** draw(TurtleGraphics.Pen p){

p.move(x1,y1);

p.move(x2,y2);

p.move(xPos,yPos);

}

**public** **void** move(**double** xAmount, **double** yAmount)

{

**double** changeX =xAmount - xPos;

**double** changeY = yAmount - yPos;

**super**.move(xAmount,yAmount);

x1 +=changeX;

x2 += changeX;

y2+= changeY;

y1+= changeY;

}

**public** **void** stretchBy(**double** factor) {

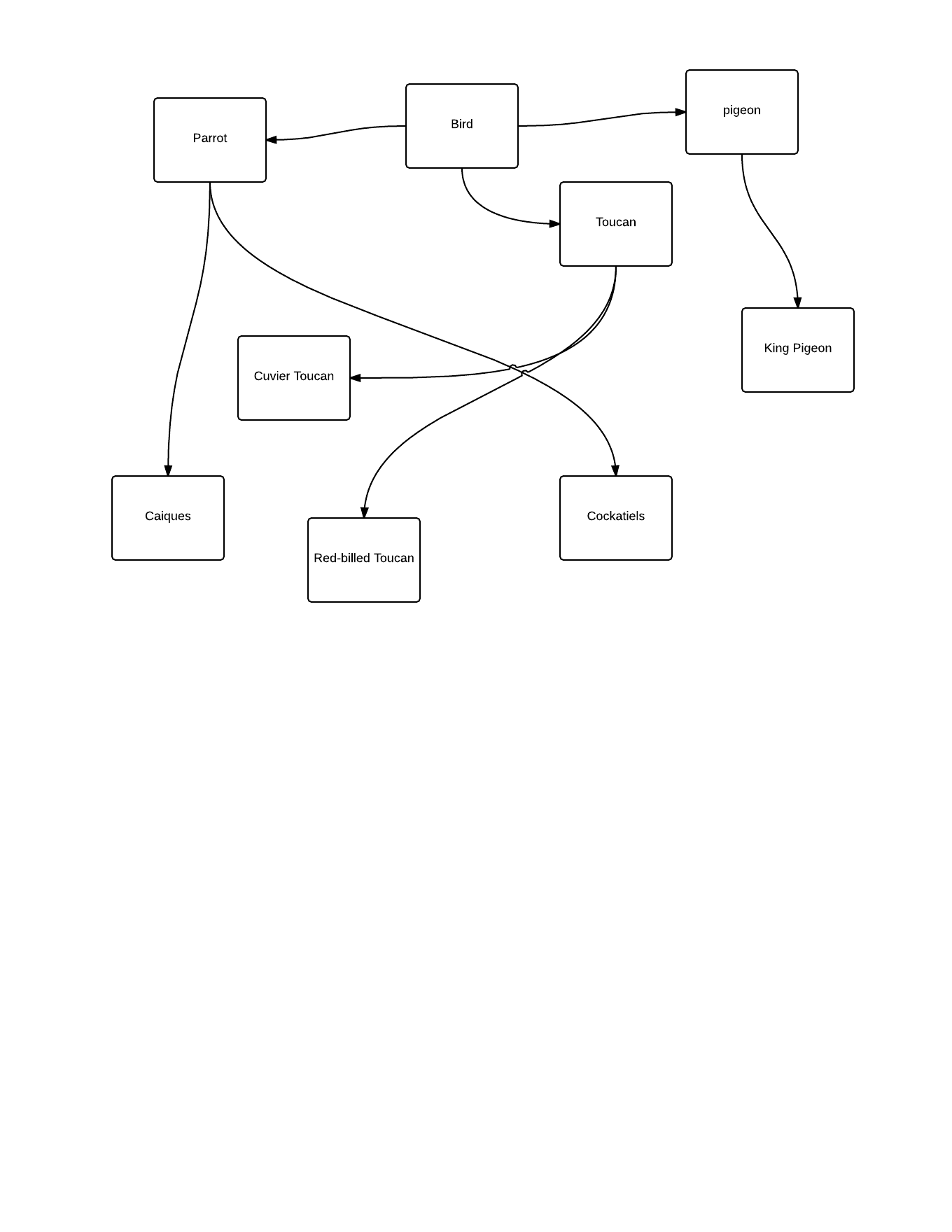
x1 \*=factor;

x2 \*=factor;

y1 \*=factor;

y2 \*=factor;

}

}// end of class

// snowman (abstractShape.java)

**private** **int** height;

**private** **int** width;

**private** **int** xSpeed;

**private** **int** ySpeed;

**private** Color color;

**public** AbstractShape(**int** x, **int** y, **int** w, **int** h){

xPos = x;

yPos = y;

width = w;

color = Color.WHITE;

xSpeed = 0;

ySpeed= 1;

}

**public** AbstractShape(**int** x, **int** y, **int** w, **int** h, Color col)

{

xPos = x;

yPos = y;

width = w;

height = h;

color = col;

xSpeed = 0;

ySpeed= 1;

}

**public** AbstractShape(**int** x, **int** y, **int** w, **int** h, Color col, **int** xSpd, **int** ySpd)

{

xPos = x;

yPos = y;

width = w;

height = h;

color = col;

xSpeed = xSpd;

ySpeed= ySpd;

}

**public** **int** getHeight()

{

**return** height;

}

**public** **int** getWidth()

{

**return** width;

}

**public** Color getColor()

{

**return** color;

}

**public** **int** getXSpeed()

{

**return** xSpeed;

}

**public** **int** getYSpeed()

{

**return** ySpeed;

}

**public** **void** setXSpeed( **int** a)

{

xSpeed = a;

}

**public** **void** setYSpeed(**int** a)

{

ySpeed = a;

}

**public** **void** setXPos( **int** a)

{

xPos = a;

}

**public** **void** setYPos( **int** a)

{

yPos = a;

}

//Name - Josh Martin

//Date -

//Class -

//Lab -Snow Man

**import** java.awt.Color;

**import** java.awt.Graphics;

**import** java.awt.Polygon;

**public** **class** SnowMan **extends** AbstractShape

{

**public** SnowMan(**int** x, **int** y, **int** w, **int** h )

{

**super**(x, y, w, h, Color.WHITE, 0, 0);

}

**public** **void** draw(Graphics window)

{ window.setColor(Color.white);

window.fillOval(450,450,150,90);

window.fillOval(475,415,100,60);

window.fillOval(490,380,70,60);

}

**public** **void** moveAndDraw(Graphics window)

{

draw(window);

}

}

//Name -Josh Martin

//Date -

//Class -AP Computer Science

//Lab -

**import** java.awt.Color;

**import** java.awt.Graphics;

**import** java.awt.Graphics2D;

**import** java.awt.Canvas;

**import** java.awt.Font;

**import** javax.swing.JPanel;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Random;

**public** **class** WinterScenePanel **extends** JPanel **implements** Runnable

{

**private** AbstractShape[] shapes;

**private** AbstractShape sMan;

**public** WinterScenePanel()

{

setVisible(**true**);

//instantiate the array to hold 50 AbstractShape references

shapes = **new** AbstractShape[50];

//populate the array with 50 unique snowflakes

Random math = **new** Random();

**for** ( **int** i= 0; i<shapes.length; i++)

{

**int** x = math.nextInt(800);

shapes[i] = **new** FancySnowFlake(x,10,10,10);

}

//instantiate a snowman

sMan = **new** SnowMan(20, 20, 40, 50);

**new** Thread(**this**).start();

}

**public** **void** update(Graphics window)

{

paint(window);

}

**public** **void** paint(Graphics window)

{

window.setColor(Color.BLUE);

window.fillRect(0,0,getWidth(), getHeight());

window.setColor(Color.WHITE);

window.drawRect(20,20,getWidth()-40,getHeight()-40);

window.setFont(**new** Font("TAHOMA",Font.BOLD,18));

window.drawString("MAKE A WINTER SCENE!",40,40);

sMan.draw(window);

//make the snowflakes appear and move down the screen

**for**(**int** i = 0; i < 50; i++)

{

shapes[i].moveAndDraw(window);

**if** (shapes[i].getYPos() > getHeight())

{

shapes[i].setYPos(0);

}

}

//check to see if any of the snowflakes need to be reset to the top of the screen

}

**public** **void** run()

{

**try**

{

**while**(**true**)

{

Thread.*currentThread*().*sleep*(35);

repaint();

}

}**catch**(Exception e)

{

}

}

}

//Name -Josh Martin

//Date -

//Class -Ap Computer Science

//Lab -

**import** java.awt.Color;

**import** java.awt.Graphics;

**import** java.awt.Polygon;

**import** java.util.Random;

//extend the AbstractShape class to make a FancySnowFlake class

**public** **class** FancySnowFlake **extends** AbstractShape

{

**public** FancySnowFlake(**int** x, **int** y, **int** w, **int** h )

{

**super**(x, y, w, h, Color.WHITE, 0, 1);

}

**public** **void** draw(Graphics window) {

window.setColor(Color.white);

window.drawOval(getXPos(),getYPos(),getWidth(),getHeight());

}

**public** **void** moveAndDraw(Graphics window) {

Random r = **new** Random();

setXPos(getXPos() +(getXSpeed()));

setYPos(getYPos()+(getXSpeed()+r.nextInt(2)));

draw(window);

**if** (getXPos() > 800 + getWidth()){

setXPos(0);

draw(window);

}

**if** (getYPos() > 650)

{

setYPos(0);

setXPos(r.nextInt(850));

}

}

}