Programming in Java 2004 Paper 1 Question 9 (AFB)

This is a full executable solution. It will be possible to obtain full marks without including all detail here, but of course the main design components must all be present, and in appropriate syntax.

```
import java.awt.event.MouseMotionListener;
import java.awt.event.MouseEvent;
import java.util.Random;

/**
 * Java conker game
 */

public class JavaConk extends BasicGame
{
    private Conker myConk;
    private Conker yourConk;
    private int breakageThreshold;

    /**
     * Constructor for Java Conker game
     */
    public JavaConk (int breakThreshold)
    {
        super();
        myConk = new Conker(30, 20, 20, 0, 0, "blue");
        yourConk = new Conker(30, 80, 20, -1, 3, "green");
        breakageThreshold = breakThreshold;

        myConk.draw();
        yourConk.draw();
    }

    public void mouseDragged(MouseEvent e) {
        myConk.updateVelocity(e.getX(), e.getY());
    }

    public void mouseMoved(MouseEvent e) {
        myConk.updateVelocity(e.getX(), e.getY());
    }

    public void play()
    {
        int i = 500;
        while (i > 0) {
            int myEnergy = (int) myConk.energy();
            int yourEnergy = (int) yourConk.energy();
            if (myConk.collidesWith(yourConk) && (myEnergy + yourEnergy) > breakageTh
                Random r = new Random();
                int myChance = r.nextInt(myEnergy);
                int yourChance = r.nextInt(yourEnergy);
                if (myChance > yourChance) {
                    yourConk.smash();
                } else {
                    myConk.smash();
                }
                yourConk.bounceOff(myConk);
            } else {
                myConk.animate();
                yourConk.animate();
            }
            i -= 1;
        }
        myConk.stopInPlace();
    }

}



/**
 * Example solution: Conker class.
 */
public class Conker extends Ball
{
    private float xVelocity;
    private float yVelocity;

    /**
     * Constructor for objects of class Conker
     */
    public Conker(int d, int x, int y, int vx, int vy, String col)
    {
        // initialise instance variables
        super(d, x, y, col);
        xVelocity = vx;
        yVelocity = vy;
```

```
}

public void updateVelocity(int newX, int newY)
{
    xVelocity = newX - getX();
    yVelocity = newY - getY();
    try {
        moveTo(newX, newY);
    } catch (OutOfBoundsException e) {
        smash();
    }
}

public float energy()
{
    float totalVelocity = (xVelocity + yVelocity);
    return totalVelocity * totalVelocity;
}

public void animate()
{
    int newX = getX() + (int) xVelocity;
    int newY = getY() + (int) yVelocity;
    try {
        moveTo(newX, newY);
    } catch(OutOfBoundsException e) {
        if (e.overHorizontalBoundary()) {
            xVelocity = -xVelocity;
            newX = e.getXboundary();
        } else {
            yVelocity = -yVelocity;
            newY = e.getYboundary();
        }
    }
   draw();
}

public void bounceOff(Conker other)
{
    int newX = (int)(Math.abs(xVelocity) + Math.abs(other.xVelocity));
    int newY = (int)(Math.abs(yVelocity) + Math.abs(other.yVelocity));
    if (other.getX() < this.getX()) {
        xVelocity = newX;
    } else {
        xVelocity = -newX;
        }
        if (other.getY() < this.getY()) {
            yVelocity = newY;
        } else {
            yVelocity = -newY;
        }
        animate();
    }
}


public class OutOfBoundsException extends Exception
{
    private String whichAxis;
    private int alternateX;
    private int alternateY;

    /**
     * Constructor for objects of class OutOfBoundsException
     */
    public OutOfBoundsException(String axis, int x, int y)
    {
        // initialise instance variables
        whichAxis = axis;
        alternateX = x;
        alternateY = y;
    }

    public boolean overHorizontalBoundary()
    {
        return (whichAxis == "x");
    }

    public int getXboundary()
    {
        return alternateX;
    }

    public int getYboundary()
    {
        return alternateY;
    }
}
```