19. Inheritance :: *Shape* Class Implementation

```
//*************************************************************************
// CLASS: Shape (Shape.java)
//*************************************************************************

/**
 * Represents a Shape.
 */
public class Shape {
    private int mX;
    private int mY;

    /**
     * Shape default ctor. Initialize the mX and mY data members to 0 and 0.
     * /
    public Shape() {
        this(0, 0);
    }

    /**
     * Shape ctor. Initialize the mX and mY data members to pX and pY.
     */
    public Shape(int pX, int pY) {
        setX(pX);
        setY(pY);
    }
```

19. Inheritance :: *Shape* Class Implementation (continued)

```
/**
 * Accessor method for the mX data member.
 */
public int getX() {
    return mX;
}

/**
 * Accessor method for the mY data member.
 */
public int getY() {
    return mY;
}

/**
 * Moves this Shape to a new location.
 */
public void move(int pNewX, int pNewY) {
    setX(pNewX);
    setY(pNewY);
}
```

19. Inheritance :: *Shape* Class Implementation (continued)

```
    /**
     * Mutator method for the mX data member.
     */
    public void setX(int pNewX) {
        mX = pNewX;
    }

    /**
     * Mutator method for the mY data member.
     */
    public void setY(int pNewY) {
        mY = pNewY;
    }
}
```