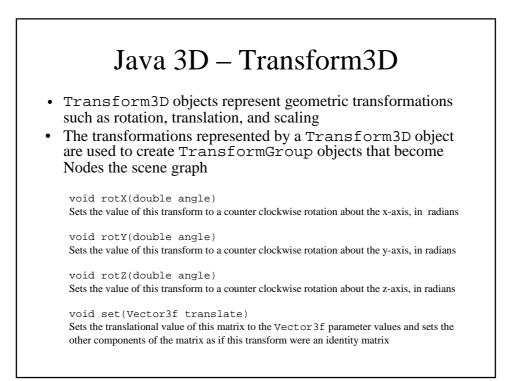
## Java 3D Transformations

• The Java 3D model for 4 x 4 transformations is

```
[ m00 m01 m02 m03 ] [x ] [x']
[ m10 m11 m12 m13 ] . [y ] = [y']
[ m20 m21 m22 m23 ] [z ] [z']
[ m30 m31 m32 m33 ] [w ] [w']
x' = m00 .x + m01 .y + m02 .z + m03 .w
y' = m10 .x + m11 .y + m12 .z + m13 .w
z' = m20 .x + m21 .y + m22 .z + m23 .w
w' = m30 .x + m31 .y + m32 .z + m33 .w
```



## Some Transform3D Methods

Transform3D()

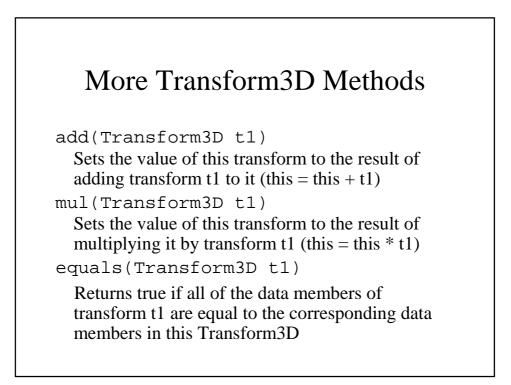
Constructs and initializes a transform to the identity matrix Transform3D(double[] matrix)

Constructs and initializes a transform from the double precision array of length 16; the top row of the matrix is initialized to the first four elements of the array, and so on

Transform3D(Matrix3f m1, Vector3f t1,float s)

Constructs and initializes a transform from the rotation matrix, translation, and scale values

Transform3D(Quat4d q1, Vector3d t1, double s) Constructs and initializes a transform from the quaternion, translation, and scale values



## Java 3D – Transform Groups

- Transform3D objects define affine homogeneous transformations but are not nodes in the scene graph
- Transform3D objects are linked into the scene graph via TransformGroup nodes
- TransformGroup objects possess a capability bit which permits certain modifications to be made after they have been made *live* (added to the scene graph) or *compiled*
- TransformGroup objects also possess a behaviour bit which permits animation, etc.

