MatrixMathematics.java

```
   package ogl.tp.service;

   import ogl.tp.exception.NoSquareException;
   import ogl.tp.model.Matrix;

   public class MatrixMathematics {

       /**
        * This class a matrix utility class and cannot be instantiated.
        */



       /**
        * Determinant of a square matrix
        * The following function find the determinant in a recursively.
        * @param matrix
        * @return
        * @throws NoSquareException
        */
       public static double determinant(Matrix matrix) throws NoSquareException {
1          if (!matrix.isSquare())
               throw new NoSquareException("matrix need to be square.");
1          if (matrix.size() == 1){
1              return matrix.getValueAt(0, 0);
           }

1          if (matrix.size()==2) {
4              return (matrix.getValueAt(0, 0) * matrix.getValueAt(1, 1)) - ( matrix.getValueAt(0, 1) * matrix.getValueAt(1, 0));
           }
           double sum = 0.0;
2          for (int i=0; i<matrix.getNcols(); i++) {
3              sum += changeSign(i) * matrix.getValueAt(0, i) * determinant(createSubMatrix(matrix, 0, i));
           }
1          return sum;
       }

       public static Matrix createSubMatrix(Matrix matrix, int excluding_row, int excluding_col) {
2          Matrix mat = new Matrix(matrix.getNrows()-1, matrix.getNcols()-1);
           int r = -1;
3          for (int i=0;i<matrix.getNrows();i++) {
1              if (i==excluding_row)
                   continue;
1              r++;
               int c = -1;
2              for (int j=0;j<matrix.getNcols();j++) {
1                  if (j==excluding_col)
                       continue;
2                  mat.setValueAt(r, ++c, matrix.getValueAt(i, j));
               }
           }
1          return mat;
       }

       /**
        * Determine the sign; i.e. even numbers have sign + and odds -
        * @param i
        * @return
        */
       private static int changeSign(int i) {
2          if (i%2==0)
1              return 1;
1          return -1;
       }
       /**
        * Creates a submatrix excluding the given row and column
        * @param matrix
        * @param excluding_row
        * @param excluding_col
        * @return
        */

       /**
        * The cofactor of a matrix
        * @param matrix
        * @return
        * @throws NoSquareException
        */
    /*   public static Matrix cofactor(Matrix matrix) throws NoSquareException {
           Matrix mat = new Matrix(matrix.getNrows(), matrix.getNcols());
           for (int i=0;i<matrix.getNrows();i++) {
               for (int j=0; j<matrix.getNcols();j++) {
```

```
        mat.setValueAt(i, j, changeSign(i) * changeSign(j) * determinant(createSubMatrix(matrix, i, j)));
      }
    }

    return mat;
  }*/
/*   public static Matrix transpose(Matrix matrix) {
    Matrix transposedMatrix = new Matrix(matrix.getNcols(), matrix.getNrows());
    for (int i=0;i<matrix.getNrows();i++) {
      for (int j=0;j<matrix.getNcols();j++) {
        transposedMatrix.setValueAt(j, i, matrix.getValueAt(i, j));
      }
    }
    return transposedMatrix;
  }*/

/*   public static Matrix inverse(Matrix matrix) throws NoSquareException {
    return (transpose(cofactor(matrix)).multiplyByConstant(1.0/determinant(matrix)));
  }*/
}
```

Mutations

```
22  1. negated conditional → KILLED
24  1. negated conditional → KILLED
25  1. replaced double return with 0.0d for ogl/tp/service/MatrixMathematics::determinant → KILLED
28  1. negated conditional → KILLED
29  1. Replaced double multiplication with division → KILLED
    2. Replaced double multiplication with division → KILLED
    3. Replaced double subtraction with addition → KILLED
    4. replaced double return with 0.0d for ogl/tp/service/MatrixMathematics::determinant → KILLED
32  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
33  1. Replaced double multiplication with division → KILLED
    2. Replaced double multiplication with division → KILLED
    3. Replaced double addition with subtraction → KILLED
35  1. replaced double return with 0.0d for ogl/tp/service/MatrixMathematics::determinant → KILLED
39  1. Replaced integer subtraction with addition → KILLED
    2. Replaced integer subtraction with addition → KILLED
41  1. changed conditional boundary → KILLED
    2. Changed increment from 1 to -1 → KILLED
    3. negated conditional → KILLED
42  1. negated conditional → KILLED
44  1. Changed increment from 1 to -1 → KILLED
46  1. changed conditional boundary → KILLED
    2. negated conditional → KILLED
47  1. negated conditional → KILLED
49  1. Changed increment from 1 to -1 → KILLED
    2. removed call to ogl/tp/model/Matrix::setValueAt → KILLED
52  1. replaced return value with null for ogl/tp/service/MatrixMathematics::createSubMatrix → KILLED
61  1. Replaced integer modulus with multiplication → KILLED
    2. negated conditional → KILLED
62  1. replaced int return with 0 for ogl/tp/service/MatrixMathematics::changeSign → KILLED
63  1. replaced int return with 0 for ogl/tp/service/MatrixMathematics::changeSign → KILLED
```

Active mutators

- CONDITIONALS_BOUNDARY
- EMPTY_RETURNS
- FALSE_RETURNS
- INCREMENTS
- INVERT_NEGS
- MATH
- NEGATE_CONDITIONALS
- NULL_RETURNS
- PRIMITIVE_RETURNS
- TRUE_RETURNS
- VOID_METHOD_CALLS

Tests examined

- ogl.tp.MatrixMathematicsTest.DeterminantTest1(ogl.tp.MatrixMathematicsTest) (1 ms)
- ogl.tp.MatrixMathematicsTest.DeterminantTest4(ogl.tp.MatrixMathematicsTest) (1 ms)
- ogl.tp.MatrixMathematicsTest.DeterminantTest3(ogl.tp.MatrixMathematicsTest) (0 ms)
- ogl.tp.MatrixMathematicsTest.DeterminantTest2(ogl.tp.MatrixMathematicsTest) (1 ms)

Report generated by PIT 1.7.2