Practice 1：Matrix Interface & Exception

An m×n matrix is a two-dimensional structure with m rows and n columns.

Matrices have two special structures, vector and square matrix. The vector can be divided into a 1×n row vector and an n×1 column vector; a square matrix is an n×n matrix. The basic operations on two matrices are addition, subtraction, and multiplication. To add (A+B) and subtract (A-B) two matrices A and B, the two matrices must have the same number of rows and columns. To multiply two matrices (A×B), the number of columns of A must be equal to the number of rows of B.

Use the Java programming language to define an **interface** for a two-dimensional structure, this interface has four abstract methods addition, subtraction, multiplication, and matrix transposition; and use object-oriented class **inheritance** to define a concrete class Matrix (matrices) and three sub-classes VectorRow (row vectors), VectorCol (column vectors), and SMatrix (square matrix). Assume that the value of each matrix element is a positive floating point number less than 1.0, and use a random number generator to generate the values of the matrix elements.

The **inner product** of two row vectors or two column vectors of equal length is defined as the products of two pairwise vector elements and then added together. Let U and V be two row or column vectors of length n, the inner product of U and V is $U \bullet V = \sum_{\mathrm{i}-0}^{n-1} U_i \times V_i$.

For an *n*×*n* matrix A, calculate its **determinant** |A|. The recursive definition of the determinant, expanded by row *i*, is:

$$|A| = \begin{cases} a_{0,0} & \text{if } n = 1 \\ \sum_{j=0}^{n-1} (-1)^{i+j} a_{i,j} \times |cofactor(A,i,j)| & \text{if } n > 1' \end{cases}$$

cofactor(A,i,j) is the (n-1)×(n-1) square matrix after removing the i-th row and j-th column of A.

At the same time, define a **matrix exception**, MatrixException, if the operand(s) of addition, subtraction, multiplication, matrix transposition, vector inner product, and square matrix determinant do not meet the requirements, then raise, propagate, and handle matrix exceptions.

In the application program, perform the following steps, and process any exception :

1. Declare and create three matrices A, B, and C of size 6×4, 6×4, and 4×6, respectively. Test expressions A+B, A-B, A×C, C×A, $(B-A)^T$, and C-A. Print the matrices A, B, and C and the resulting matrices of the above expressions.
2. Declare row vector vR. Set each of row 0 to row 6 of matrix A to vR and print each row.
3. Declare row vector vC. Set each of column 0 to column 4 of matrix A to vC and print each column.
4. Compute and print $A_{*,0} \bullet B_{*,0}$, $A_{*,0} \times C_{0,*}$, and $A_{*,1} \bullet C_{*,1}$.
5. Declare and create square matrix S. Compute and print |S|, |B×C|, and |A|.

Output Sample:

```
Matrix A:
  0.2802   0.6970   0.4332   0.1415
  0.5550   0.9290   0.2165   0.0582
  0.8805   0.6117   0.7398   0.5087
  0.0421   0.4831   0.2239   0.8715
  0.6559   0.4685   0.7221   0.8865
  0.8673   0.5710   0.2086   0.0393
======================
Matrix B:
  0.1545   0.5785   0.0450   0.7648
  0.2362   0.8213   0.9246   0.2637
  0.7309   0.0908   0.0549   0.2336
  0.1323   0.2217   0.7717   0.3573
  0.8741   0.7771   0.0041   0.9629
  0.7397   0.7328   0.0058   0.8692
======================
Matrix C:
  0.6406   0.0536   0.5528   0.0698   0.8336   0.5778
  0.3704   0.5679   0.9789   0.6805   0.8975   0.3748
  0.0471   0.7755   0.6213   0.4104   0.9794   0.8317
  0.1235   0.8029   0.1751   0.7315   0.6679   0.8760
======================
Matrix A+B:
  0.4347   1.2755   0.4781   0.9063
  0.7913   1.7503   1.1411   0.3219
  1.6114   0.7024   0.7947   0.7422
  0.1744   0.7049   0.9956   1.2288
  1.5299   1.2456   0.7262   1.8494
  1.6070   1.3038   0.2145   0.9085
======================
Matrix A-B:
  0.1257   0.1185   0.3882  -0.6233
  0.3188   0.1077  -0.7081  -0.2055
  0.1496   0.5209   0.6849   0.2751
 -0.0902   0.2614  -0.5478   0.5141
 -0.2182  -0.3087   0.7180  -0.0764
```

```
  0.1276  -0.1618   0.2028  -0.8300
======================
Matrix A*C:
  0.4756   0.8604   1.1311   0.7752   1.3780   0.9074
  0.7170   0.7720   1.3610   0.8024   1.5475   0.9000
  0.8882   1.3767   1.6342   1.1534   2.3473   1.7990
  0.3241   1.1499   0.7880   1.0611   1.2701   1.1551
  0.7371   1.5729   1.4250   1.3094   2.2665   1.9317
  0.7818   0.5641   1.1750   0.5635   1.4661   0.9231
======================
Matrix C*B:
  1.6810   1.5516   0.1694   1.9631
  2.0586   1.8926   1.1266   2.0949
  2.1702   2.1822   1.0789   2.1983
  1.6653   2.0699   1.3299   2.0130
======================
Matrix (B-A)^T:
 -0.1257  -0.3188  -0.1496   0.0902   0.2182  -0.1276
 -0.1185  -0.1077  -0.5209  -0.2614   0.3087   0.1618
 -0.3882   0.7081  -0.6849   0.5478  -0.7180  -0.2028
  0.6233   0.2055  -0.2751  -0.5141   0.0764   0.8300
======================
>>>> Catch matrix operation exceptions.
matrixInterface.MatrixException: Subtraction: mismatch matrix size.
    at matrixInterface.Matrix.minus(Matrix.java:97)
    at matrixInterface.MatrixApp.main(MatrixApp.java:43)
****************************
Row 0 of matrix A:
  0.2802   0.6970   0.4332   0.1415
======================
Row 1 of matrix A:
  0.5550   0.9290   0.2165   0.0582
======================
Row 2 of matrix A:
  0.8805   0.6117   0.7398   0.5087
======================
Row 3 of matrix A:
  0.0421   0.4831   0.2239   0.8715
```

```
======================
Row 4 of matrix A:
  0.6559   0.4685   0.7221   0.8865
======================
Row 5 of matrix A:
  0.8673   0.5710   0.2086   0.0393
======================
>>>> Catch row vector exception.
matrixInterface.MatrixException: Row 6 does not exist.
     at matrixInterface.VectorRow.getVector(VectorRow.java:25)
     at matrixInterface.MatrixApp.main(MatrixApp.java:56)
****************************
Column 0 of matrix A:
  0.2802
  0.5550
  0.8805
  0.0421
  0.6559
  0.8673
======================
Column 1 of matrix A:
  0.6970
  0.9290
  0.6117
  0.4831
  0.4685
  0.5710
======================
Column 2 of matrix A:
  0.4332
  0.2165
  0.7398
  0.2239
  0.7221
  0.2086
======================
Column 3 of matrix A:
  0.1415
```

```
  0.0582
  0.5087
  0.8715
  0.8865
  0.0393
======================
>>>> Catch column vector exception.
matrixInterface.MatrixException: Column 4 does not exist.
     at matrixInterface.VectorCol.getVector(VectorCol.java:25)
     at matrixInterface.MatrixApp.main(MatrixApp.java:72)
****************************
Row A[0][*]:
  0.2802   0.6970   0.4332   0.1415
Row B[0][*]:
  0.1545   0.5785   0.0450   0.7648
Inner product of A[0][*] and B[0][*]: 0.5742
======================
Column C[*][0]:
  0.6406
  0.3704
  0.0471
  0.1235
A[0][*]*C[*][0]:
  0.4756
======================
Row A[1][*]:
  0.5550   0.9290   0.2165   0.0582
Row C[1][*]:
  0.3704   0.5679   0.9789   0.6805   0.8975   0.3748
>>>> Catch inner product row vector length mismatch exception.
matrixInterface.MatrixException: The two row vectors are not of the same length.
     at matrixInterface.VectorRow.innerProduct(VectorRow.java:60)
     at matrixInterface.MatrixApp.main(MatrixApp.java:107)
****************************
Column A[*][0]:
  0.2802
  0.5550
  0.8805
```

```
  0.0421
  0.6559
  0.8673
Column B[*][0]:
  0.1545
  0.2362
  0.7309
  0.1323
  0.8741
  0.7397
Inner product of A[*][0] and B[*][0]: 2.0383
======================
Row C[0][*]:
  0.6406   0.0536   0.5528   0.0698   0.8336   0.5778
A[*][0]*[0]C[*]:
  0.1795   0.0150   0.1549   0.0196   0.2336   0.1619
  0.3555   0.0298   0.3068   0.0387   0.4627   0.3207
  0.5640   0.0472   0.4867   0.0615   0.7340   0.5088
  0.0270   0.0023   0.0233   0.0029   0.0351   0.0243
  0.4201   0.0352   0.3626   0.0458   0.5467   0.3790
  0.5556   0.0465   0.4795   0.0605   0.7230   0.5012
======================
Column A[*][1]:
  0.6970
  0.9290
  0.6117
  0.4831
  0.4685
  0.5710
Column C[*][1]:
  0.0536
  0.5679
  0.7755
  0.8029
>>>> Catch inner product column vector length mismatch exception.
matrixInterface.MatrixException: The two column vectors are of not the same
length.
     at matrixInterface.VectorCol.innerProduct(VectorCol.java:60)
```

```
    at matrixInterface.MatrixApp.main(MatrixApp.java:138)
*****************************
Matrix S:
  0.3350   0.4757   0.6278   0.9323   0.9263
  0.9563   0.3035   0.7937   0.5485   0.2581
  0.7123   0.9454   0.7815   0.2240   0.7469
  0.0423   0.6708   0.3294   0.6923   0.4853
  0.8710   0.7403   0.4683   0.7519   0.6059
The determinant of square matrix S: 1.4691E-01
======================
Matrix B*C:
  0.4098   0.9857   0.8136   0.9824   1.2028   1.0134
  0.5317   1.4078   1.5553   1.1478   2.0158   1.4444
  0.5332   0.3209   0.5680   0.3062   0.9006   0.7066
  0.2473   1.0184   0.8322   0.7382   1.3038   1.1145
  0.9668   1.2645   1.4151   1.2959   2.0733   1.6433
  0.8529   1.1582   1.2821   1.1886   1.8606   1.4684
The determinant of square matrix B*C: 1.9023E-17
======================
Matrix A:
  0.2802   0.6970   0.4332   0.1415
  0.5550   0.9290   0.2165   0.0582
  0.8805   0.6117   0.7398   0.5087
  0.0421   0.4831   0.2239   0.8715
  0.6559   0.4685   0.7221   0.8865
  0.8673   0.5710   0.2086   0.0393
>>>> Catches the determinant square exception.
matrixInterface.MatrixException: Determinant: Not a square matrix.
    at matrixInterface.Matrix.determinant(Matrix.java:144)
    at matrixInterface.MatrixApp.main(MatrixApp.java:161)
*****************************
```