

디지털컨버전스 기반 UI/UX Front 전문 개발자양성 과정(비트캠프)

강사 : 이승훈
수강생 : 오진욱

Dimensional Array

다차원 배열

- 개념 : 수학의 행렬과 같은 구조, 2차원 배열, 3차원 배열 등 n차원 배열이 존재
- 작성법 : `int [][] arr = {{1, 2, 3, 4, 5}};` `int [][] arr = new int [a][b];`
- a=행/b=열을 뜻함
- // 이중 배열의 경우 안쪽에 있는 녀석이 큰 공간이다.
- // 바깥쪽에 있는 배열은 큰 공간을 다시 나눈 작은 공간이다.

Example

- 2중 for문을 이용하여 행 / 열의 값을 구한다
- 행과 열은 0부터 시작한다.
- i, j = 행(i), 열(j)의 길이로 설정
- 사진 참조
- $[0][0] = 1$, $[0][1] = 0$, $[0][2] = 0$

```
public class DoubleArrayTest01 {
    public static void main(String[] args) {

        //int[][] arr = new int[3][3];
        int[][] arr = {
            {1, 0, 0},
            {0, 1, 0},
            {0, 0, 1}
        };

        //   [0] [1] [2]
        // -----
        // | 1 | 0 | 0 |   [0]
        // -----
        // | 0 | 1 | 0 |   [1]
        // -----
        // | 0 | 0 | 1 |   [2]
        // -----

        for(int i = 0; i < 3; i++) {
            for(int j = 0; j < 3; j++) {
                // 이중 배열의 경우 안쪽에 있는 녀석이 큰 공간이다.
                // 바깥쪽에 있는 배열은 큰 공간을 다시 나눈 작은 공간이다.
                System.out.printf("%2d", arr[i][j]);
            }
            System.out.println("");
        }
    }
}
```

```
System.out.println("new를 통해 Heap에 할당하기");

int[][] mat = new int[3][3];

for(int i = 0; i < 3; i++) {
    for(int j = 0; j < 3; j++) {
        mat[i][j] = (i + 1) * (j + 1);
        System.out.printf("%2d", mat[i][j]);
    }
    System.out.println("");
}
}
```

Exercise

Quiz- 10일차 수업

1) 오늘 만든 Matrix 클래스를 개조하여사용자의 배열을 입력받아 행렬을 만들도록 개조해보자!

실제 matlab의 풀 라이선스 가격은수십억 ~ 백억 이상 되는 소프트웨어다.

ex) numpy나 matlab이라는 파이썬 라이브러리혹은 소프트웨어에 존재하는 기능으로아래와 같은 것이 되게 만드는 것이다.

```
int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9}; Matrix A = new Matrix(arr, 3, 3);
```

혹은 Matrix A = new Matrix(arr, 3);둘 다 동작이 되어야 할 것이다.

2) 행렬 두 개를 만들고 이 두 개의 행렬 덧셈을 수행하게 만들어보자!행렬의 차원은 편한대로 설정해도 무방하다.

3) 덧셈이 있다면 뺄셈도 있으니 뺄셈을 만들어보자!

4) 그렇다면 곱셈이 빠질 수 없으므로 곱셈도 만든다.


```

104 private boolean checkAvailable(int[] arr, int row, int col) { 1중 배열, 2중 배열, 3중 배열.
105     int len = arr.length;
106
107     boolean res = (len == row * col ? true : false);
108
109     if(res) {
110         this.row = row;
111         this.col = col;
112     }
113
114     return res;
115 }
116
117 private boolean checkAvailable(int[] arr, int num) { 이중 배열, 2중 배열, 3중 배열.
118     int row = arr.length;
119     int col = arr[0].length;
120     int len = row * col;
121     // System.out.printf("[ ] row = %d, col = %d\n", row, col);
122
123     /*
124     if(len % row == 0) {
125         this.row = row;
126         this.col = len / row;
127     } else {
128         System.out.printf("행렬로 변환할 수 없습니다.\n");
129         System.out.printf("올바른 차원을 입력하세요.\n");
130         System.out.printf("혹은 적절한 숫자(행)를 입력하세요\n");
131         return false;
132     }
133     */
134
135     return checkDivideElement(len, num);
136 }
137
138 private boolean checkAvailable(int[] arr, int row) {
139     int len = arr.length;
140
141     /*
142     if(len % row == 0) {
143         this.row = row;
144         this.col = len / row;
145     } else {
146         System.out.printf("행렬로 변환할 수 없습니다.\n");
147         System.out.printf("올바른 차원을 입력하세요.\n");
148         System.out.printf("혹은 적절한 숫자(행)를 입력하세요\n");
149         return false;
150     }
151     */
152
153     return checkDivideElement(len, row);
154 }

```

checkDivideElement Class가 같은 것.

이중 배열, 2중 배열, 3중 배열. → 2중 num을 쓰지? row와 같은 것만?

같은 것만 있는 것이지만 class를 만들지?

```

156 private boolean checkDimension(Matrix mat) {
157     int row = mat.getRow();
158     int col = mat.getCol();
159     return (this.row == row) && (this.col == col);
160 }
161
162 private boolean checkDimension(Matrix A, Matrix B) {
163     int Arow = A.getRow();
164     int Brow = B.getRow();
165     int Acol = A.getCol();
166     int Bcol = B.getCol();
167
168     return (Arow == Brow) && (Acol == Bcol);
169 }
170
171 public void addMatrix(Matrix mat) {
172     if(checkDimension(mat)) {
173         int[][] srcMat = mat.getMat();
174
175         for (int i = 0; i < row; i++) {
176             for (int j = 0; j < col; j++) {
177                 this.mat[i][j] = this.mat[i][j] + srcMat[i][j];
178             }
179         }
180     }
181 }
182
183 public void addMatrix(Matrix A, Matrix B) {
184     if(checkDimension(A, B)) {
185         int[][] matA = A.getMat();
186         int[][] matB = B.getMat();
187
188         for (int i = 0; i < row; i++) {
189             for (int j = 0; j < col; j++) {
190                 mat[i][j] = matA[i][j] + matB[i][j];
191             }
192         }
193     }
194 }
195
196 public void subMatrix(Matrix mat) {
197     if(checkDimension(mat)) {
198         int[][] srcMat = mat.getMat();
199
200         for (int i = 0; i < row; i++) {
201             for (int j = 0; j < col; j++) {
202                 this.mat[i][j] = this.mat[i][j] - srcMat[i][j];
203             }
204         }
205     }
206 }
207 }

```

Matrix mat? → Matrix class를 사용한다. ex) Matrix A/B/C.

이전 클래스의 값은 가져와서 더하기. getter 사용.

위: → 무슨 뜻인가?

아래: →

더하는 메소드. (A=A+B)

Private mat? Private mat?

더하는 메소드 (R=A+B)

이제 method 하나만 필요함.

19:37 Wed 13 Jan

github.com

```

209 public void subMatrix(Matrix A, Matrix B) {
210     if(checkDimension(A, B)) {
211         int[][] matA = A.getMat();
212         int[][] matB = B.getMat();
213
214         for (int i = 0; i < row; i++) {
215             for (int j = 0; j < col; j++) {
216                 mat[i][j] = matA[i][j] - matB[i][j];
217             }
218         }
219     }
220 }

```

3by3 * 3by2 = OK. → 3by2

```

224 public boolean checkMulDimension(Matrix A, Matrix B) {
225     int Brow = B.getRow();
226     int Acol = A.getCol();
227
228     return (Brow == Acol);
229 }

```



```

231 public void mulMatrix(Matrix A, Matrix B) {
232     if(checkMulDimension(A, B)) {
233         int[][] matA = A.getMat();
234         int[][] matB = B.getMat();
235
236         // 00  01  02      00  01  02
237         // 10  11  12      10  11  12
238         // 20  21  22      20  21  22
239         // 00 * 00 + 01 * 10 + 02 * 20: [0][0]
240         // 00 * 01 + 01 * 11 + 02 * 21: [0][1]
241         // 00 * 02 + 01 * 12 + 02 * 22: [0][2]
242
243         // 10 * 00 + 11 * 10 + 12 * 20: [1][0]
244         // 10 * 01 + 11 * 11 + 12 * 21: [1][1]
245         // 10 * 02 + 11 * 12 + 12 * 22: [1][2]
246
247         // 20 * 00 + 21 * 10 + 22 * 20: [2][0]
248         // 20 * 01 + 21 * 11 + 22 * 21: [2][1]
249         // 20 * 02 + 21 * 12 + 22 * 22: [2][2]
250         mat[0][0] = matA[0][0] * matB[0][0] +
251             matA[0][1] * matB[1][0] +
252             matA[0][2] * matB[2][0];
253         mat[0][1] = matA[0][0] * matB[0][1] +
254             matA[0][1] * matB[1][1] +
255             matA[0][2] * matB[2][1];
256         mat[0][2] = matA[0][0] * matB[0][2] +
257             matA[0][1] * matB[1][2] +
258             matA[0][2] * matB[2][2];
259         // 10 * 00 + 11 * 10 + 12 * 20: [1][0]
260         // 10 * 01 + 11 * 11 + 12 * 21: [1][1]
261         // 10 * 02 + 11 * 12 + 12 * 22: [1][2]
262         mat[1][0] = matA[1][0] * matB[0][0] +
263             matA[1][1] * matB[1][0] +
264             matA[1][2] * matB[2][0];
265         mat[1][1] = matA[1][0] * matB[0][1] +
266             matA[1][1] * matB[1][1] +
267             matA[1][2] * matB[2][1];
268         mat[1][2] = matA[1][0] * matB[0][2] +
269             matA[1][1] * matB[1][2] +
270             matA[1][2] * matB[2][2];
271         // 20 * 00 + 21 * 10 + 22 * 20: [2][0]
272         // 20 * 01 + 21 * 11 + 22 * 21: [2][1]
273         // 20 * 02 + 21 * 12 + 22 * 22: [2][2]
274         mat[2][0] = matA[2][0] * matB[0][0] +
275             matA[2][1] * matB[1][0] +
276             matA[2][2] * matB[2][0];
277         mat[2][1] = matA[2][0] * matB[0][1] +
278             matA[2][1] * matB[1][1] +
279             matA[2][2] * matB[2][1];
280         mat[2][2] = matA[2][0] * matB[0][2] +
281             matA[2][1] * matB[1][2] +
282             matA[2][2] * matB[2][2];
283     }
284 }
285

```

하드코딩!
for문보다 속도↑

```

286 public void allocRandomMatrix() {
287     for(int i = 0; i < row; i++) {
288         for(int j = 0; j < col; j++) {
289             mat[i][j] = (int)(Math.random() * 10);
290         }
291     }
292 }
293
294 public int getRow() {
295     return row;
296 }
297
298 public int getCol() {
299     return col;
300 }
301
302 public int[][] getMat() {
303     return mat;
304 }
305
306 // n by n 행렬의 판별식
307 // ex) 3 by 3
308 // 1    2    3
309 // 4    5    6 =====>
310 // 7    8    9
311 //
312 // 1 * {(5 * 9) - (6 * 8)} +
313 // 2 * {(6 * 7) - (4 * 9)} +
314 // 3 * {(4 * 8) - (5 * 7)}
315 // 이 결과가 0 이 아니면 역행렬이 존재한다.
316
317 public void printMatrix() {
318     for(int i = 0; i < row; i++) {
319         for(int j = 0; j < col; j++) {
320             System.out.printf("%4d", mat[i][j]);
321         }
322         System.out.println("");
323     }
324 }
325

```

int[row][col] array
에 0~10까지 랜덤할듯.

다행/비행/공통한
getter!

또 class/method
구현을 위한 메소드!

```
import Tenth.Matrix;
public class HomeworkReview {
    public static void main(String[] args) {
        int[] arr1 = {1, 2, 3, 4, 5, 6, 7, 8, 9};
        int[][] arr2 = {
            {1, 2, 3},
            {2, 4, 6},
            {3, 6, 9}
        };
        int[][] arr3 = {
            {1, 2, 3, 4},
            {2, 4, 6, 8},
            {3, 6, 9, 12}
        };

        Matrix wrong = new Matrix(arr1, row: 3);
        wrong.allocRandomMatrix();
        wrong.printMatrix();

        System.out.println("Matrix A:");
        Matrix A = new Matrix(arr1, row: 3, col: 3);
        A.printMatrix();
    }
}
```

```
Matrix wrong = new Matrix(arr1, row: 3);
wrong.allocRandomMatrix();
wrong.printMatrix();

System.out.println("Matrix A:");
Matrix A = new Matrix(arr1, row: 3, col: 3);
A.printMatrix();

System.out.println("Matrix B:");
Matrix B = new Matrix(arr2, row: 3);
B.printMatrix();

System.out.println("Matrix C:");
Matrix C = new Matrix(arr3, row: 4);
C.printMatrix();
```

```
System.out.println("R = A + B");
Matrix R = new Matrix(row: 3, col: 3);
R.addMatrix(A, B);
R.printMatrix();

System.out.println("A = A + B");
A.addMatrix(B);
A.printMatrix();

System.out.println("R = A - B");
R.subMatrix(A, B);
R.printMatrix();

System.out.println("A = A - B");
A.subMatrix(B);
A.printMatrix();

System.out.println("R = A * B");
R.mulMatrix(A, B);
R.printMatrix();
```

행렬로 변환할 수 없습니다.
올바른 차원을 입력하세요.
혹은 적절한 숫자(행)를 입력하세요

Matrix A:

A 처리 테스트

```
1 2 3
4 5 6
7 8 9
```

Matrix B:

```
1 2 3
2 4 6
3 6 9
```

Matrix C:

```
1 2 3
4 2 4
6 8 3
6 9 12
```

R = A + B

```
2 4 6
6 9 12
10 14 18
```

A = A + B

```
2 4 6
6 9 12
10 14 18
```

R = A - B

```
1 2 3
4 5 6
7 8 9
```

A = A - B

```
1 2 3
4 5 6
7 8 9
```

R = A * B

```
14 28 38
32 64 96
50 100 150
```