Eric Roberts
CS 106A

Handout #41
May 9, 2012

Multidimensional Arrays

Multidimensional Arrays

Eric Roberts
CS 106A
May 9, 2012

Multidimensional Arrays

- Because the elements of an array can be of any Java type, those elements can themselves be arrays. Arrays of arrays are called ***multidimensional arrays***.
- In Java, you can create a multidimensional array by using multiple brackets in both the type and the initialization parts of the declaration. For example, you can create array space for a 3×3 tic-tac-toe board using the following declaration:

```
char[][] board = new char[3][3];
```

- This declaration creates a two-dimensional array of characters that is organized like this:

board[0][0]	board[0][1]	board[0][2]
board[1][0]	board[1][1]	board[1][2]
board[2][0]	board[2][1]	board[2][2]

Arrays of Arrays

- Internally, Java represents multidimensional arrays as ***arrays of arrays.*** It is often important to keep this fact in mind when you are creating array structures.
- In Java, it is often necessary to initialize the rows of a two-dimensional array individually, as in the following revised code for creating the tic-tac-toe board:

```
char[][] board = new char[3][];
for (int i = 0; i < 3; i++) {
   board[i] = new char[3];
}
```

Static Initialization

- Java makes it easy to initialize the elements of an array as part of a declaration. The syntax is

```
type[] name = {elements};
```

where *elements* is a list of the elements of the array separated by commas. The length of the array is automatically set to be the number of values in the list.

- For example, the following declaration initializes the variable `powersOfTen` to the values 10^0, 10^1, 10^2, 10^3, and 10^4:

```
int[] powersOfTen = {1, 10, 100, 1000, 10000};
```

This declaration creates an integer array of length 5 and initializes the elements as specified.

Initializing Multidimensional Arrays

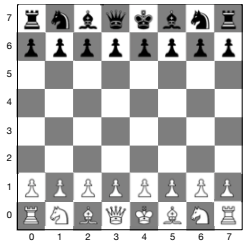
- You can initialize a multidimensional array when you declare it by using nested braces to reflect the levels of array nesting.
- For example, you can declare and initialize a multiplication table for the digits 0 to 9 like this:

```
private static final int[][] MULTIPLICATION_TABLE = {
   {  0,  0,  0,  0,  0,  0,  0,  0,  0,  0 },
   {  0,  1,  2,  3,  4,  5,  6,  7,  8,  9 },
   {  0,  2,  4,  6,  8, 10, 12, 14, 16, 18 },
   {  0,  3,  6,  9, 12, 15, 18, 21, 24, 27 },
   {  0,  4,  8, 12, 16, 20, 24, 28, 32, 36 },
   {  0,  5, 10, 15, 20, 25, 30, 35, 40, 45 },
   {  0,  6, 12, 18, 24, 30, 36, 42, 48, 56 },
   {  0,  7, 14, 21, 28, 35, 42, 49, 56, 63 },
   {  0,  8, 16, 24, 32, 40, 48, 56, 64, 72 },
   {  0,  9, 18, 27, 36, 45, 54, 63, 72, 81 }
};
```

Exercise: Dreaming up Applications

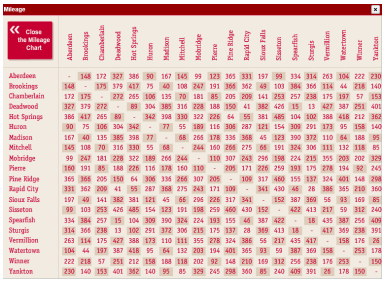
- The plan for today is to go through a series of data structure design exercises and then pick one or two to embellish into larger programs.
- Your first exercise, however, is to spend a few minutes thinking about real-world examples of multidimensional arrays other than the ones that appear on the subsequent slides. Use your imagination!

Exercise: Representing a Chess Board



Exercise: Mileage Chart

I was intrigued to discover that the top Google Images hit for "mileage chart" was the following table for South Dakota:



Mileage

Close the Mileage Chart

	Aberdeen	Brookings	Chamberlain	Deadwood	Hot Springs	Huron	Madison	Mitchell	Mobridge	Pierre	Pine Ridge	Rapid City	Sioux Falls	Sisseton	Spearfish	Sturgis	Vermillion	Watertown	Winner	Yankton
Aberdeen	-	148	172	327	386	90	167	145	99	123	165	331	197	99	334	314	263	104	222	230
Brookings	148	-	175	379	417	75	40	108	247	191	366	362	49	103	384	366	314	44	218	140
Chamberlain	172	175	-	272	265	106	135	70	181	85	205	209	141	253	257	238	175	197	57	153
Deadwood	327	379	272	-	89	304	385	314	228	188	350	41	382	429	15	13	427	387	251	401
Hot Springs	386	417	265	89	-	342	398	330	322	226	64	55	381	485	104	102	388	418	212	362
Huron	90	75	106	304	342	-	77	55	189	116	306	287	121	154	306	291	173	95	158	140
Madison	167	40	135	385	398	77	-	68	266	178	336	366	45	123	390	372	110	64	188	95
Mitchell	145	108	70	314	330	55	68	-	244	160	266	275	66	191	324	306	111	112	118	85
Mobridge	99	247	181	228	322	189	266	244	-	110	307	243	296	198	224	215	355	203	202	329
Pierre	160	191	85	188	226	116	178	160	110	-	205	171	226	259	193	175	278	194	92	245
Pine Ridge	165	366	205	350	64	306	336	266	307	205	-	109	317	400	155	137	324	401	148	298
Rapid City	331	362	209	41	55	287	368	275	243	171	109	-	341	430	46	28	386	365	210	360
Sioux Falls	197	49	141	382	381	121	45	66	296	226	317	341	-	152	387	369	56	93	169	85
Sisseton	99	103	253	426	485	154	123	191	198	259	400	430	152	-	422	413	217	59	312	240
Spearfish	334	384	257	15	104	306	390	324	224	193	155	46	387	422	-	18	435	387	256	409
Sturgis	314	366	238	13	102	291	372	306	235	175	137	28	369	413	18	-	417	369	238	391
Vermillion	263	114	175	427	388	173	110	111	355	278	324	386	56	217	435	417	-	158	176	28
Watertown	104	44	197	387	418	95	64	132	203	194	401	365	93	59	387	369	158	-	253	178
Winner	222	218	57	251	212	158	188	118	202	92	148	210	169	312	256	238	176	253	-	150
Yankton	230	140	153	401	362	140	95	85	329	245	298	360	85	240	409	391	26	178	150	-

Exercise: Sudoku

Design a program that checks legality of a Sudoku position, in which each digit between 1 and 9 must appear exactly once in each row, column, and 3x3 square.

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

Exercise: Crossword Numbering

What structure might you use to represent the black squares on a crossword puzzle grid?