159 XTENDED

A WEEKLY REVIEW

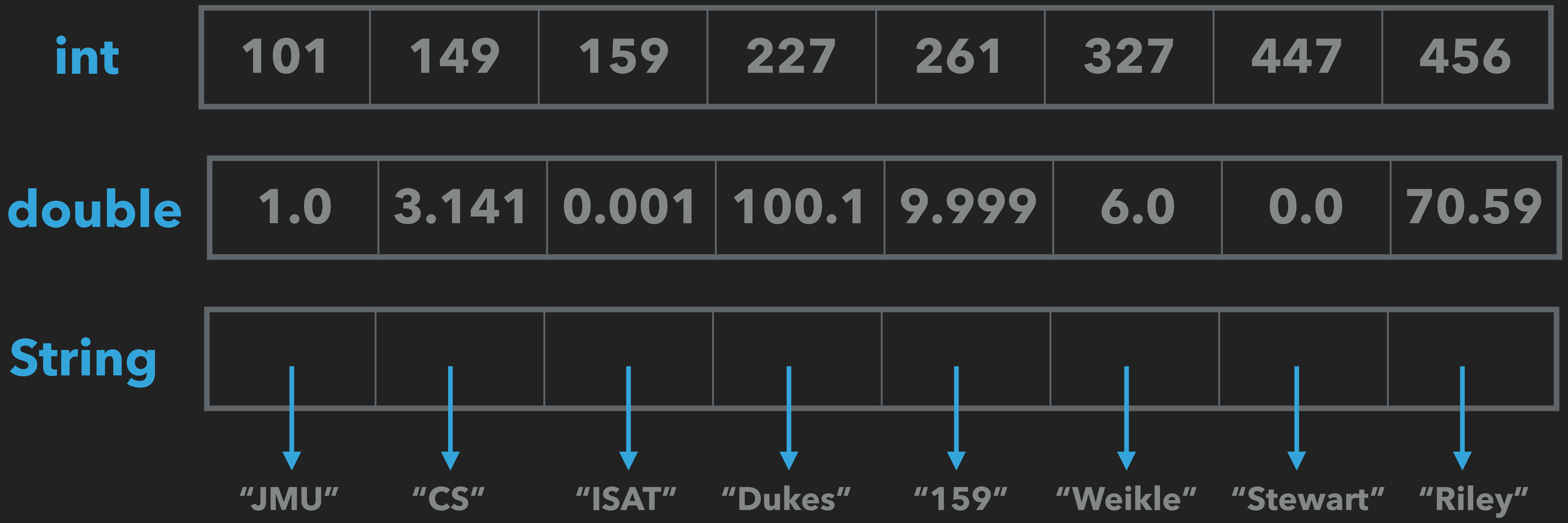
WEEK 2 REVIEW, 2D ARRAYS

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

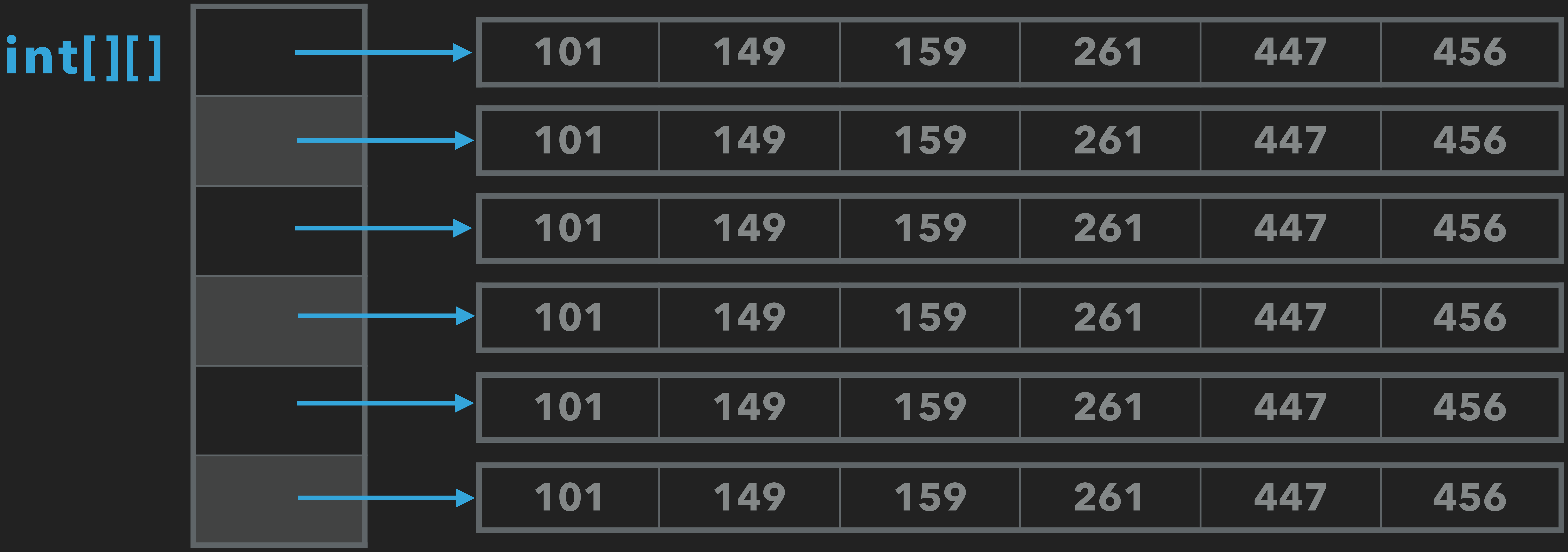
Arrays can hold many different data types in java.



int	101	149	159	227	261	327	447	456
double	1.0	3.141	0.001	100.1	9.999	6.0	0.0	70.59
String	"JMU"	"CS"	"ISAT"	"Dukes"	"159"	"Weikle"	"Stewart"	"Riley"

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

They can also hold other arrays. Arrays are reference types like Strings.
A 2D array is nothing more than an array holding references to other arrays.



101	149	159	261	447	456
101	149	159	261	447	456
101	149	159	261	447	456
101	149	159	261	447	456
101	149	159	261	447	456
101	149	159	261	447	456

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

Looking at our `gameScores` 2D array:

Home	5	1	0	10	4	5	7	3	5	10
Away	3	7	9	0	0	15	9	10	0	1

Question: What is the **type** of the expression `gameScores`?

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

Looking at our `gameScores` 2D array:

Home	5	1	0	10	4	5	7	3	5	10
Away	3	7	9	0	0	15	9	10	0	1

Question: What is the **type** of the expression `gameScores`?

A 2D array of ints.

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

Looking at our `gameScores` 2D array:

Home	5	1	0	10	4	5	7	3	5	10
Away	3	7	9	0	0	15	9	10	0	1

Question: What is the **type** of the expression `gameScores[0]`?

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

Looking at our `gameScores` 2D array:

Home	5	1	0	10	4	5	7	3	5	10
Away	**3**	**7**	**9**	**0**	**0**	**15**	**9**	**10**	**0**	**1**

Question: What is the **type** of the expression `gameScores[0]`?

A single-dimensional array of ints.

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

Looking at our `gameScores` 2D array:

Home	5	1	0	10	4	5	7	3	5	10
Away	3	7	9	0	0	15	9	10	0	1

Question: What is the **type** of the expression `gameScores[0][0]`?

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

Looking at our `gameScores` 2D array:

Home	5	1	0	10	4	5	7	3	5	10
Away	3	7	9	0	0	15	9	10	0	1

Question: What is the **type** of the expression `gameScores[0][0]`?

An int.

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: WHAT MAKES UP A 2D ARRAY?

Note: This table is from Prof. Michael Norton's notes "Arrays: Single & Multidimensional" from his CS 159 course.

Expression	Type	Description
table	int[][]	2D array of integers, or array of integer arrays
table[5]	int[]	array of integers
table[5][3]	int	integer

ACTIVITY: HLLA SCOREKEEPING

- The Harrisonburg Little League, or the HLLA, is a local league of softball, baseball, and tee ball teams that young children local to Harrisonburg and Rockingham County can compete in.
- The local little softball team, the Harrisonburg Bandits, would like a computer program that they can use to keep score during games.
- All little league teams have exactly 10 players.

2D ARRAY REVIEW

```
public class Scorekeeper
{
    int[] homeScores;
    int[] awayScores;

    public Scorekeeper()
    {
        homeScores = new int[10];
        awayScores = new int[10];
    }

    public void addHomeScore(int player, int score)
    {
        homeScores[player] = score;
    }

    public void addAwayScore(int player, int score)
    {
        awayScores[player] = score;
    }
}
```

Is it necessary to declare two separate arrays here?

ACTIVITY: HLLA SCOREKEEPING

- Declare and instantiate a 2D array called “gameScores” that will hold all of the scores for the home team and the away team.

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: HLLA SCOREKEEPING

```
int[][] gameScores = new int[2][10];
```

Key point: Rows then Columns

Home	5	1	0	10	4	5	7	3	5	10
Away	3	7	9	0	0	15	9	10	0	1

ACTIVITY: HLLA SCOREKEEPING

- Write a line of code that corrects the "addHomeScore" method to work with our 2D array.

```
public void addHomeScore(int player, int score)
{
    homeScores[player] = score;
}
```

Correct this line!

Pull up **Socrative.com**

and join room **XTEND159**

ACTIVITY: HLLA SCOREKEEPING

```
public void addHomeScore(int player, int score)
{
    gameScores[0][player] = score;

}
```

ACTIVITY: HLLA SCOREKEEPING

```
public void addHomeScore(int player, int score)
{
    gameScores[0][player] = score;

}
```

This index is referencing an individual softball player's score, which is an element in our 2D array.

This index is referencing a team's scores, which is a row in our 2D array.

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

```
public void resetToZero()
{
      . . .
}
```

```
int[][] gameScores = new int[2][10];
```

Home	5	1	0	10	4	5	7	3	5	10
Away	3	7	9	0	0	15	9	10	0	1

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

How you should solve it

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

Column by Column

How you should not solve it

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

	j		
i	1	2	3
	6	5	4
	9	8	7
	12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

j

i

0	2	3
6	5	4
9	8	7
12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player’s scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

		j	
i	0	0	3
	6	5	4
	9	8	7
	12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

			j
i	0	0	0
	6	5	4
	9	8	7
	12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

j

	0	0	0
i	0	5	4
	9	8	7
	12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

		j	
	0	0	0
i	0	0	4
	9	8	7
	12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

			j
	0	0	0
i	0	0	0
	9	8	7
	12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

j

	0	0	0
	0	0	0
i	0	8	7
	12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called resetToZero that iterates through the entirety of gameScores and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

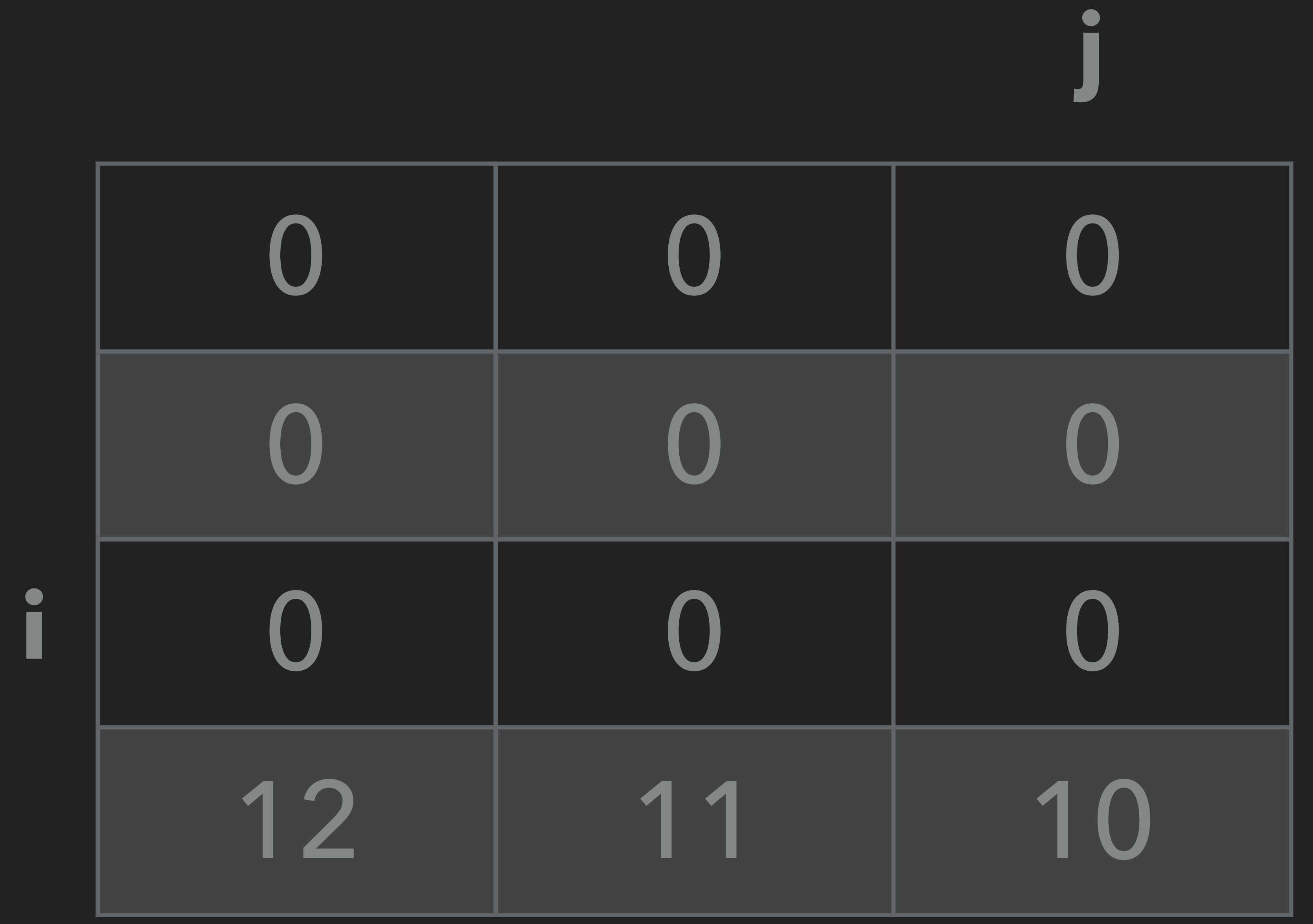
	j		
	0	0	0
	0	0	0
i	0	0	7
	12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player’s scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```



0	0	0
0	0	0
0	0	0
12	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

	j		
	0	0	0
	0	0	0
	0	0	0
i	0	11	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player’s scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

		j	
	0	0	0
	0	0	0
	0	0	0
i	0	0	10

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

Row by Row

```
public void resetToZero()
{
    for (int i = 0; i < gameScores.length; i++)
    {
        for (int j = 0; j < gameScores[i].length; j++)
        {
            gameScores[i][j] = 0;
        }
    }
}
```

			j
	0	0	0
	0	0	0
	0	0	0
i	0	0	0

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

	i		
j	1	2	3
	6	5	4
	9	8	7
	12	11	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

	i		
j	0	2	3
	6	5	4
	9	8	7
	12	11	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

i

j

0	2	3
0	5	4
9	8	7
12	11	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

	i		
	0	2	3
	0	5	4
j	0	8	7
	12	11	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

	i		
	0	2	3
	0	5	4
	0	8	7
j	0	11	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

	i		
j	0	0	3
	0	5	4
	0	8	7
	0	11	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

	i		
	0	0	3
j	0	0	4
	0	8	7
	0	11	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

		i	
	0	0	3
	0	0	4
j	0	0	7
	0	11	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

		i	
	0	0	3
	0	0	4
	0	0	7
j	0	0	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player’s scores on both teams to zero.

			i
j	0	0	0
	0	0	4
	0	0	7
	0	0	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.

			i
	0	0	0
j	0	0	0
	0	0	7
	0	0	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player’s scores on both teams to zero.

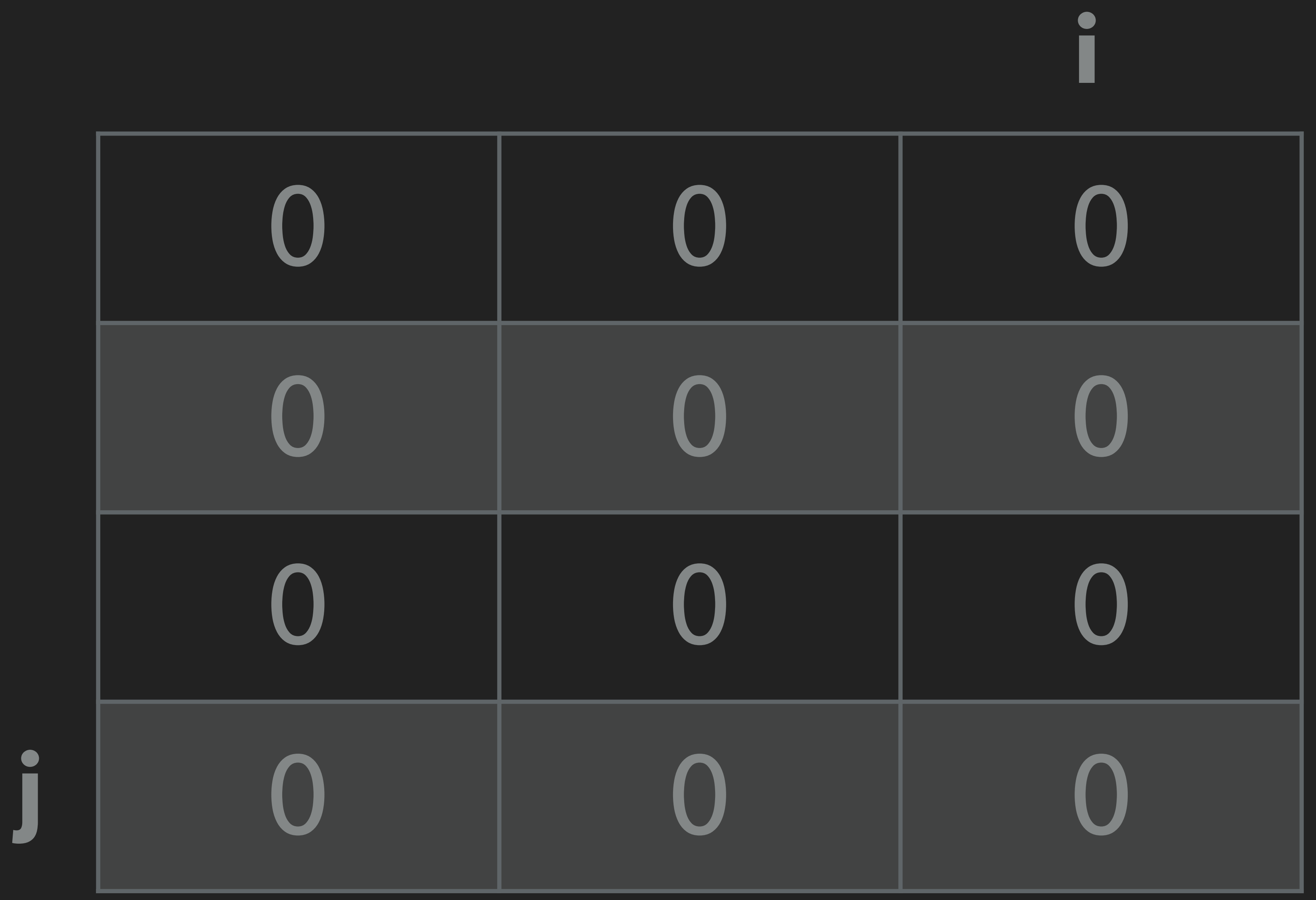
			i
	0	0	0
	0	0	0
j	0	0	0
	0	0	10

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCOREKEEPING

- Write a method called `resetToZero` that iterates through the entirety of `gameScores` and sets all player's scores on both teams to zero.



0	0	0
0	0	0
0	0	0
0	0	0

Column by Column

```
public void resetToZero()
{
    for (int i = 0; i < gameScores[0].length; i++)
    {
        for (int j = 0; j < gameScores.length; j++)
        {
            gameScores[j][i] = 0;
        }
    }
}
```

ACTIVITY: HLLA SCORE COUNTING

- The HLLA *loved* our scorekeeper and again wants our help.
- Finish the following method:

```
public static int[] findBestTeam(int[][] teamScores)
{
      . . .
}
```

- This method receives a 2D array holding each HLLA team's game scores for a given season. Each row is a team's scores for the season, and each element is a team's score in a particular game. Note: not all teams play in the same number of games.
- The method should find which team scored the most points over the entire season and return an int array holding all of their scores.

ACTIVITY: HLLA SCORE COUNTING

```
public static int[] findBestTeam(int[][] teamScores)
{
    int[] bestTeam;
    int bestTeamScore;

    for (int i = 0; i < teamScores.length; i++)
    {
        int currentTeamScore = 0;

        for (int j = 0; j < teamScores[i].length; j++)
        {
            currentTeamScore += teamScores[i][j];
        }

        if (currentTeamScore > bestTeamScore)
        {
            bestTeam = teamScores[i];
            bestTeamScore = currentTeamScore;
        }
    }

    return bestTeam;
}
```

ACTIVITY: HLLA SCORE COUNTING

- The HLLA would like to know how many total points their players scored this year so that they can ask the player's parents for more donations.
- Finish the following method:

```
public static int getTotalScore(int[][] teamScores)
{
      . . .
}
```

- This method receives a 2D array holding each HLLA team's game scores for a given season. Each row is a team's scores for the season, and each element is a team's score in a particular game. Note: not all teams play in the same number of games.
- The method should return the sum of all points scored by all teams in all games for the season.
- You should check to see if the incoming array is null or empty.

ACTIVITY: HLLA SCORE COUNTING

```
public static int getTotalScore(int[][] teamScores)
{
    int totalScore = 0;

    if (teamScores == null || teamScores.length <= 0)
    {
        return 0;
    }

    for (int i = 0; i < teamScores.length; i++)
    {
        for (int j = 0; j < teamScores[i].length; j++)
        {
            totalScore += teamScores[i][j];
        }
    }

    return totalScore;
}
```