Arrays 4/4

Outline

- Multidimensional Arrays
- Two-Dimensional Array as an Array of Arrays
- Using the **`length`** Instance Variable
- Multidimensional Array Parameters and Returned Values
- A Grade Book Class
- The Two-Dimensional Array **`grade`**
- Example

- Multidimensional Arrays ...

- It is sometimes useful to have an array with more than one index
- Multidimensional arrays are declared and created in basically the same way as one-dimensional arrays
  - You simply use as many square brackets as there are indices
  - Each index must be enclosed in its own brackets

```
double[][]table = new double[100][10];
int[][][] figure = new int[10][20][30];
Person[][] = new Person[10][100];
```

… - Multidimensional Arrays …

- Multidimensional arrays may have any number of indices, but perhaps the most common number is two
  - Two-dimensional array can be visualized as a two-dimensional display with the first index giving the row, and the second index giving the column

    ```
    char[][] a = new char[5][12];
    ```

  - Note that, like a one-dimensional array, each element of a multidimensional array is just a variable of the base type (in this case, `char`)

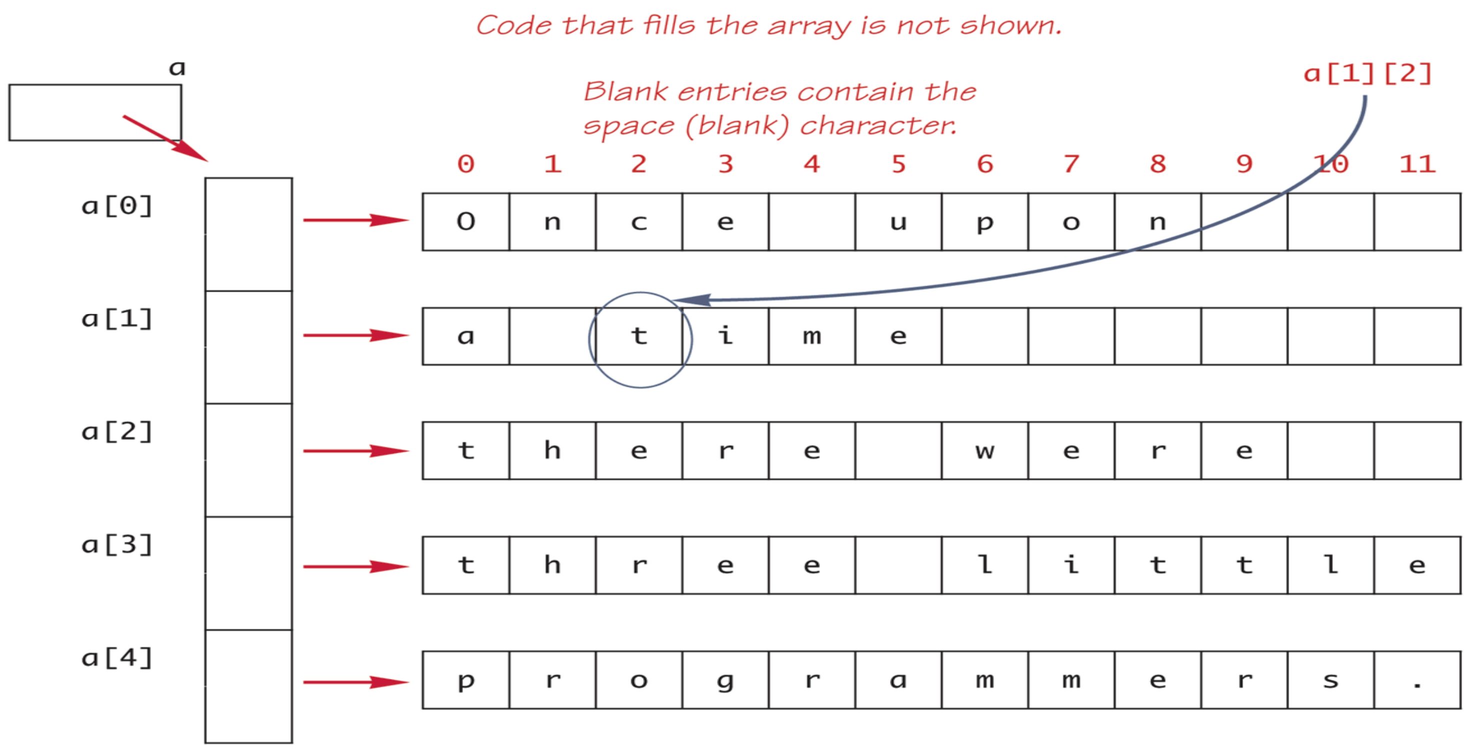
... - Multidimensional Arrays

- In Java, a two-dimensional array, such as **`a`**, is actually an array of arrays
  - The array **`a`** contains a reference to a one-dimensional array of size 5 with a base type of **`char[]`**
  - Each indexed variable (**`a[0]`**, **`a[1]`**, etc.) contains a reference to a one-dimensional array of size 12, also with a base type of **`char[]`**
- A three-dimensional array is an array of arrays of arrays, and so forth for higher dimensions

- Two-Dimensional Array as an Array of Arrays ...

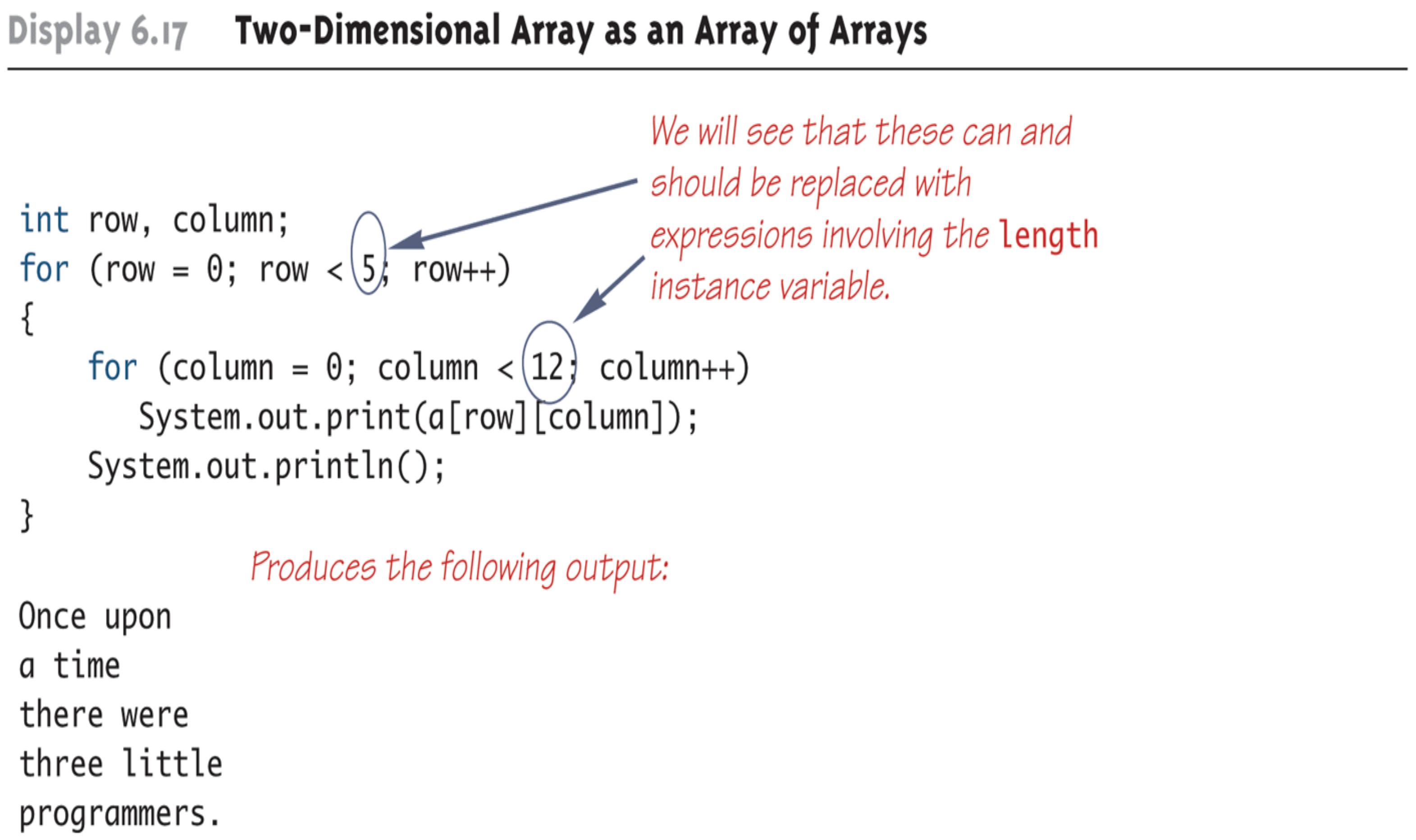
Display 6.17 **Two-Dimensional Array as an Array of Arrays**

```
char[][] a = new char[5][12];
```



(continued)

… - Two-Dimensional Array as an Array of Arrays



Display 6.17 Two-Dimensional Array as an Array of Arrays

```
int row, column;
for (row = 0; row < 5; row++)
{
    for (column = 0; column < 12; column++)
       System.out.print(a[row][column]);
    System.out.println();
}
```

We will see that these can and should be replaced with expressions involving the `length` *instance variable.*

Produces the following output:

```
Once upon
a time
there were
three little
programmers.
```

- Using the `length` Instance Variable ...

```
char[][] page = new char[30][100];
```

- The instance variable `length` does not give the total number of indexed variables in a two-dimensional array
  - Because a two-dimensional array is actually an array of arrays, the instance variable `length` gives the number of first indices (or "rows") in the array
    - `page.length` is equal to 30
  - For the same reason, the number of second indices (or "columns") for a given "row" is given by referencing `length` for that *"row" variable*
    - `page[0].length` is equal to 100

... - Using the `length` Instance Variable

- The following program demonstrates how a nested `for` loop can be used to process a two-dimensional array
  - Note how each `length` instance variable is used

```
int row, column;
for (row = 0; row < page.length; row++)
  for (column = 0; column < page[row].length;column++)
    page[row][column] = 'Z';
```

- Multidimensional Array Parameters and Returned Values ...

- Methods may have multidimensional array parameters
  - They are specified in a way similar to one-dimensional arrays
  - They use the same number of sets of square brackets as they have dimensions

```
public void myMethod(int[][] a)
{ . . . }
```

  - The parameter **a** is a two-dimensional array

... - Multidimensional Array Parameters and Returned Values

- Methods may have a multidimensional array type as their return type
  - They use the same kind of type specification as for a multidimensional array parameter

```
public double[][] aMethod()
{ . . . }
```

  - The method `aMethod` returns an array of `double`

- A Grade Book Class ...

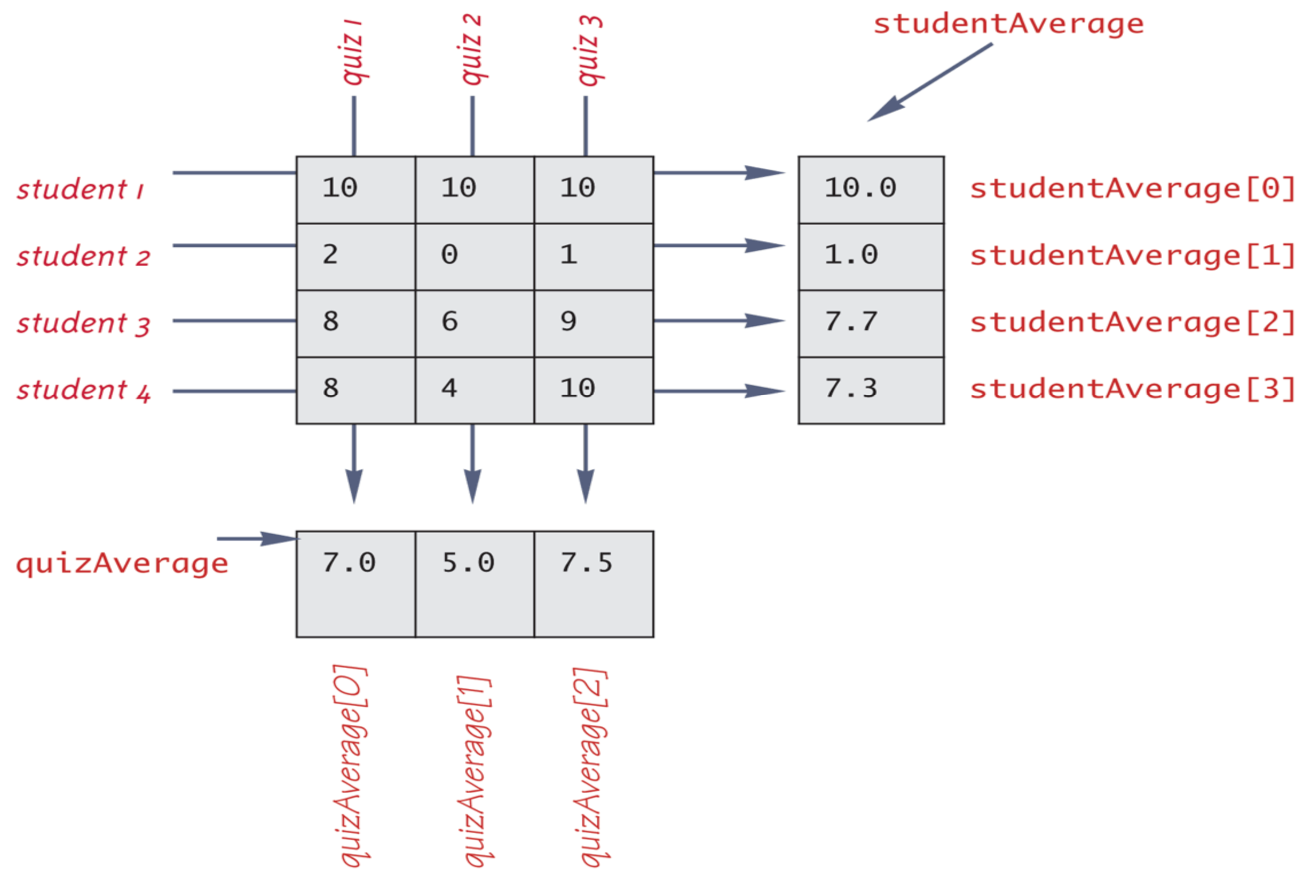
- As an example of using arrays in a program, a class **`GradeBook`** is used to process quiz scores
- Objects of this class have three instance variables
  - **`grade`**: a two-dimensional array that records the grade of each student on each quiz
  - **`studentAverage`**: an array used to record the average quiz score for each student
  - **`quizAverage`**: an array used to record the average score for each quiz

... - A Grade Book Class

- The score that student 1 received on quiz number 3 is recorded in **`grade[0][2]`**
- The average quiz grade for student 2 is recorded in **`studentAverage[1]`**
- The average score for quiz 3 is recorded in **`quizAverage[2]`**
- Note the relationship between the three arrays

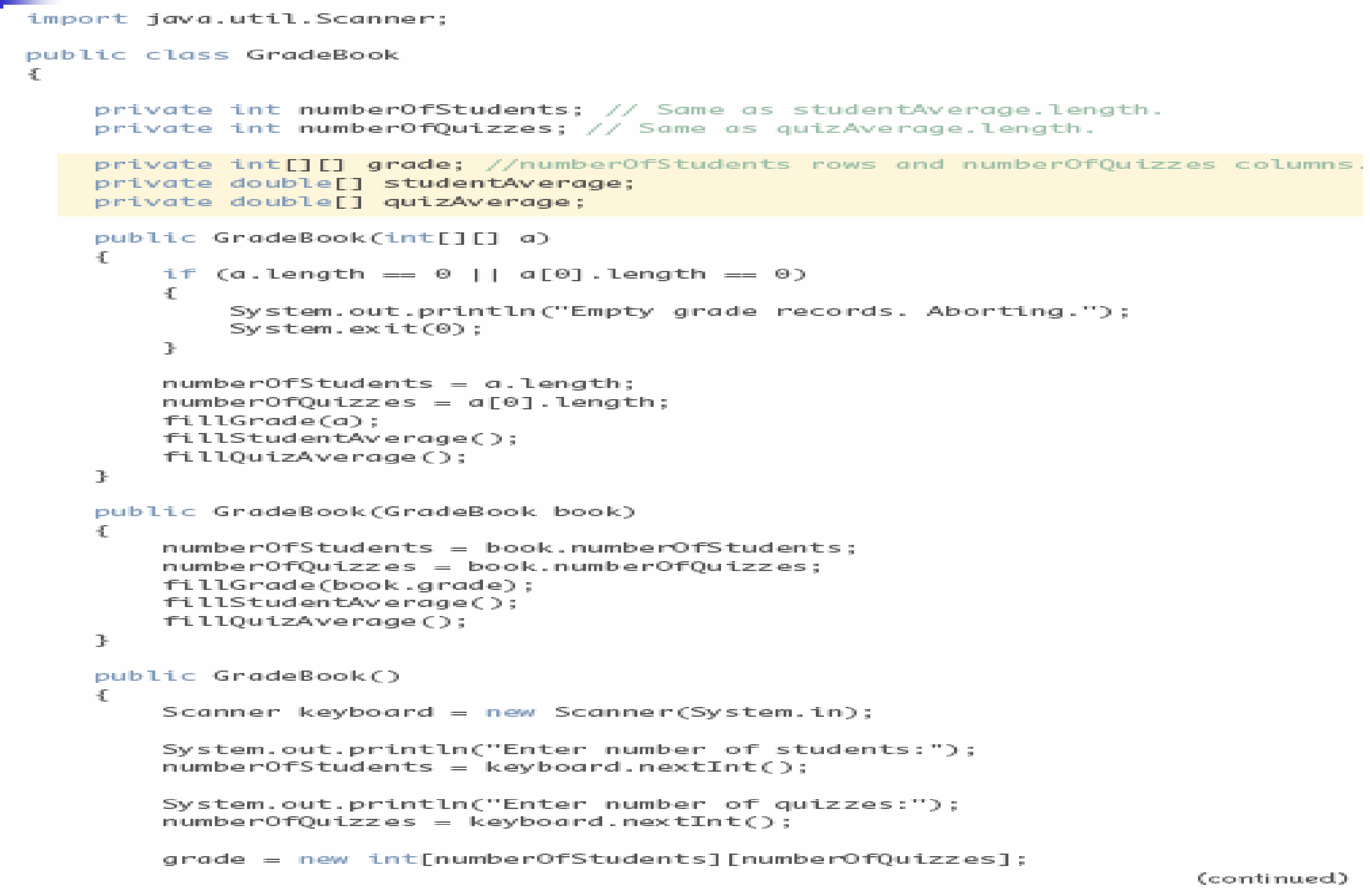
- The Two-Dimensional Array **grade**

Display 6.19 **The Two-Dimensional Array `grade`**



	quiz 1	quiz 2	quiz 3	studentAverage
student 1	10	10	10	10.0
student 2	2	0	1	1.0
student 3	8	6	9	7.7
student 4	8	4	10	7.3
quizAverage	7.0	5.0	7.5	

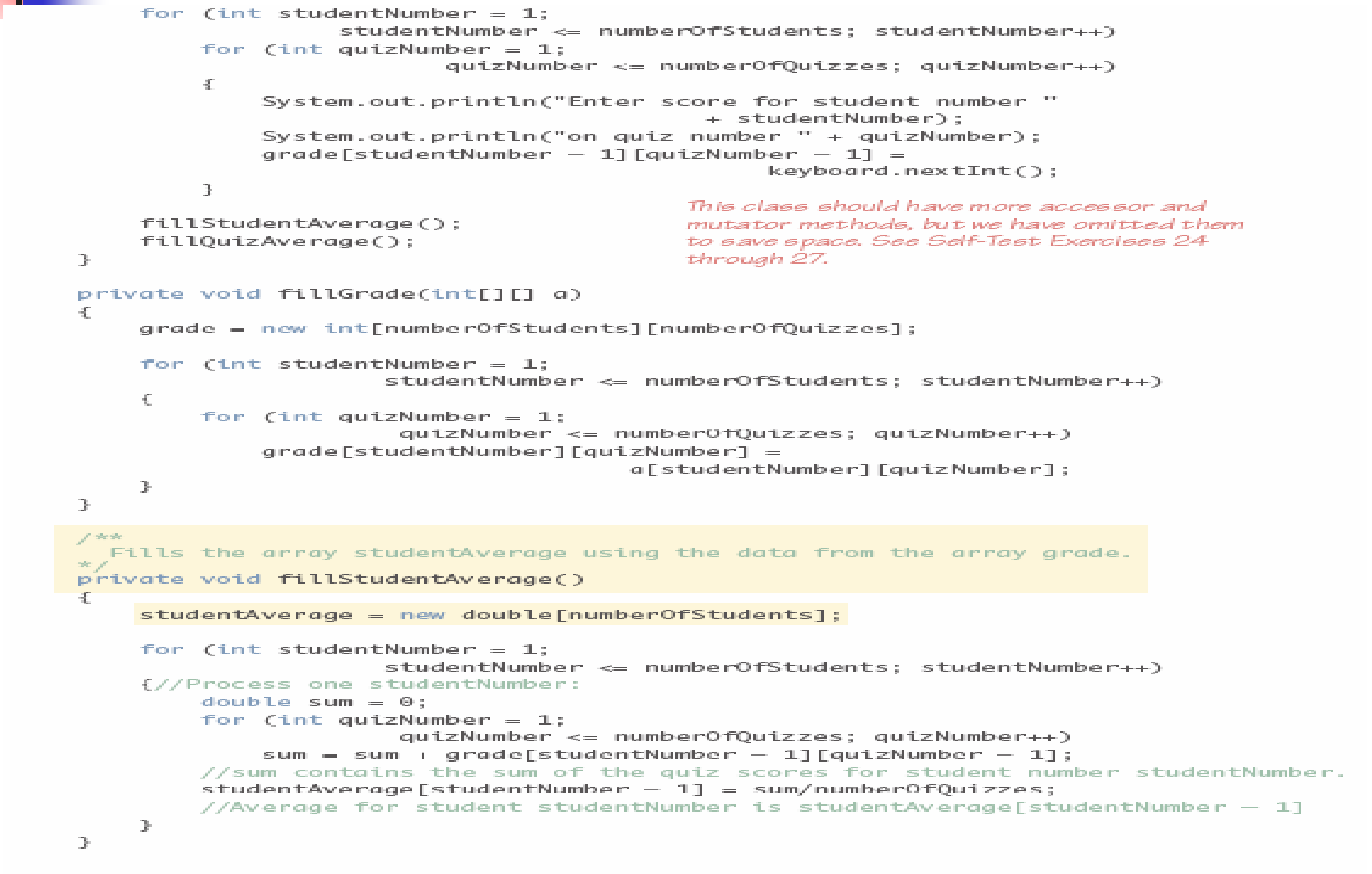
- Example ...



```
import java.util.Scanner;

public class GradeBook
{
    private int numberOfStudents; // Same as studentAverage.length.
    private int numberOfQuizzes; // Same as quizAverage.length.

    private int[][] grade; //numberOfStudents rows and numberOfQuizzes columns.
    private double[] studentAverage;
    private double[] quizAverage;

    public GradeBook(int[][] a)
    {
        if (a.length == 0 || a[0].length == 0)
        {
            System.out.println("Empty grade records. Aborting.");
            System.exit(0);
        }

        numberOfStudents = a.length;
        numberOfQuizzes = a[0].length;
        fillGrade(a);
        fillStudentAverage();
        fillQuizAverage();
    }

    public GradeBook(GradeBook book)
    {
        numberOfStudents = book.numberOfStudents;
        numberOfQuizzes = book.numberOfQuizzes;
        fillGrade(book.grade);
        fillStudentAverage();
        fillQuizAverage();
    }

    public GradeBook()
    {
        Scanner keyboard = new Scanner(System.in);

        System.out.println("Enter number of students:");
        numberOfStudents = keyboard.nextInt();

        System.out.println("Enter number of quizzes:");
        numberOfQuizzes = keyboard.nextInt();

        grade = new int[numberOfStudents][numberOfQuizzes];
```

(continued)

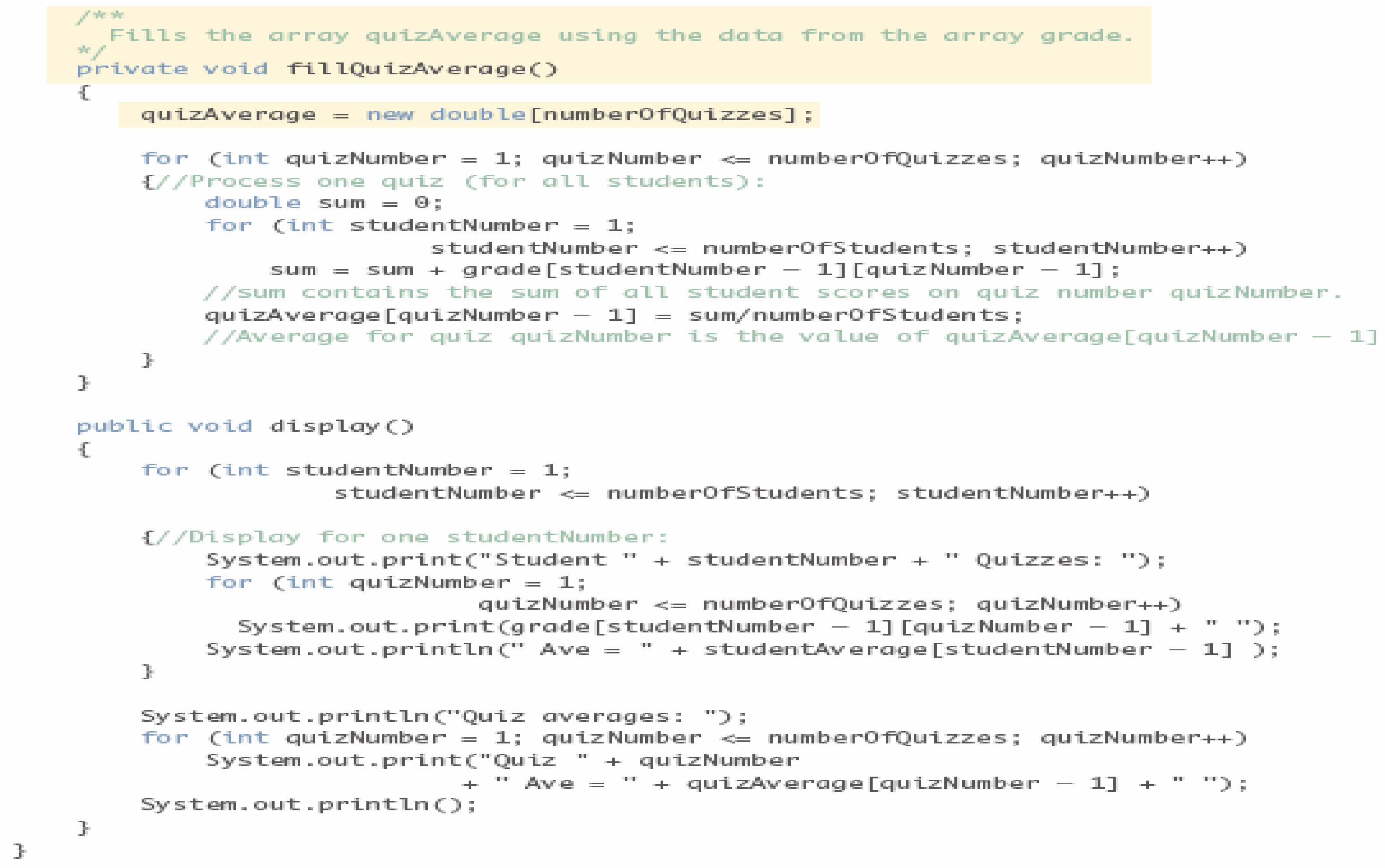
... - Example ...



```
        for (int studentNumber = 1;
                    studentNumber <= numberOfStudents; studentNumber++)
            for (int quizNumber = 1;
                        quizNumber <= numberOfQuizzes; quizNumber++)
            {
                System.out.println("Enter score for student number "
                                        + studentNumber);
                System.out.println("on quiz number " + quizNumber);
                grade[studentNumber - 1][quizNumber - 1] =
                                             keyboard.nextInt();
            }

        fillStudentAverage();
        fillQuizAverage();
    }

    private void fillGrade(int[][] a)
    {
        grade = new int[numberOfStudents][numberOfQuizzes];

        for (int studentNumber = 1;
                        studentNumber <= numberOfStudents; studentNumber++)
        {
            for (int quizNumber = 1;
                         quizNumber <= numberOfQuizzes; quizNumber++)
                grade[studentNumber][quizNumber] =
                                    a[studentNumber][quizNumber];
        }
    }

    /**
      Fills the array studentAverage using the data from the array grade.
    */
    private void fillStudentAverage()
    {
        studentAverage = new double[numberOfStudents];

        for (int studentNumber = 1;
                        studentNumber <= numberOfStudents; studentNumber++)
        {//Process one studentNumber:
            double sum = 0;
            for (int quizNumber = 1;
                         quizNumber <= numberOfQuizzes; quizNumber++)
                sum = sum + grade[studentNumber - 1][quizNumber - 1];
            //sum contains the sum of the quiz scores for student number studentNumber.
            studentAverage[studentNumber - 1] = sum/numberOfQuizzes;
            //Average for student studentNumber is studentAverage[studentNumber - 1]
        }
    }
```

This class should have more accessor and mutator methods, but we have omitted them to save space. See Self-Test Exercises 24 through 27.

... - Example



```java
    /**
     Fills the array quizAverage using the data from the array grade.
    */
    private void fillQuizAverage()
    {
        quizAverage = new double[numberOfQuizzes];

        for (int quizNumber = 1; quizNumber <= numberOfQuizzes; quizNumber++)
        {//Process one quiz (for all students):
            double sum = 0;
            for (int studentNumber = 1;
                           studentNumber <= numberOfStudents; studentNumber++)
                sum = sum + grade[studentNumber - 1][quizNumber - 1];
            //sum contains the sum of all student scores on quiz number quizNumber.
            quizAverage[quizNumber - 1] = sum/numberOfStudents;
            //Average for quiz quizNumber is the value of quizAverage[quizNumber - 1]
        }
    }

    public void display()
    {
        for (int studentNumber = 1;
                    studentNumber <= numberOfStudents; studentNumber++)

        {//Display for one studentNumber:
            System.out.print("Student " + studentNumber + " Quizzes: ");
            for (int quizNumber = 1;
                             quizNumber <= numberOfQuizzes; quizNumber++)
              System.out.print(grade[studentNumber - 1][quizNumber - 1] + " ");
            System.out.println(" Ave = " + studentAverage[studentNumber - 1] );
        }

        System.out.println("Quiz averages: ");
        for (int quizNumber = 1; quizNumber <= numberOfQuizzes; quizNumber++)
            System.out.print("Quiz " + quizNumber
                               + " Ave = " + quizAverage[quizNumber - 1] + " ");
        System.out.println();
    }
}
```

THE END