Write a Java class that will represent a PGM image. Objects of your class will be used to enhance the code we wrote in class to read and modify PGM images. Specifications for your class are provided below.

1. The name of your class should be `PGMImage`.

2. Your class should have three instance variables:
   - a 2D array of integers, representing the grayscale pixel values;
   - the number of rows in the image; and
   - the number of columns in the image.

   For instance variables, remember that you should declare them at the top of your class, but do not assign any values there. Values will be assigned inside the constructor (see below).

3. Your class should have a constructor that accepts as a parameter a 2D array of integers representing the grayscale pixel values of an image. Your constructor should:
   - Set your instance variables for rows and columns using the appropriate widths from the incoming 2D array.
   - Construct (build) your instance-variable 2D array, since you will know the number of rows and columns as specified by the 2D array given as an argument to your parameter.
   - Use a doubly-nested for loop to copy each pixel value from the argument into your newly-constructed instance-variable 2D array. Recall that an appropriate doubly-nested for loop will look similar to:

     ```java
     for (int r = 0; r < numRows; r++)
     {
         for (int c = 0; c < numCols; c++)
         {
             // copy code goes here
         }
     }
     ```

4. Your class should implement the following getter methods:
   - `int getNumRows()`
   - `int getNumCols()`
   - `int getPixelAt(int row, int col)`

5. Your class should also implement a method with the following signature:

   ```java
   public int[][] getPixelArray()
   ```

   The method should internally declare and construct a new 2D array of integers of the appropriate size, and then copy, pixel by pixel, each of the values from your instance variable 2D array into this new 2D array. At the end, return your new copy. (You would not want to just return your instance-variable 2D array. Why? You would be returning the memory address of your instance variable, which means you would be allowing the user to modify the 2D array stored inside your object, which defeats the purpose of encapsulating your data within the class using the `private` designation. Hence, you make a new copy of the 2D array and return that copy.)

As a look-ahead to Monday, once your new class is complete, we will be able to use your class in the `ImageProcessor.java` program similar to the following:

```java
public static void main(String[] args)
{
    int[][] pixels = readImageFile(args[0]); // code from Friday’s class
    PGMImage image = new PGMImage(pixels); // an instance of your class

    System.out.println(image.getNumRows());
    System.out.println(image.getNumCols());
    System.out.println(image.getPixelAt(0,0));
}
```