Multidimensional Arrays of Objects

Multidimensional arrays of primitives:

```java
cchar[ ][ ] page = new char[50][100];
```

What about multidimensional arrays of objects? Let's explore this.

**Day**: This class stores appointments for a person on a given day.
```java
private ArrayList appointments
private int dayNumber
private String name
```

**MonthlyCalendar**: This class stores each day of the current month.
```java
private String monthName
private Day[][] calendar; two-dimensional array of days for current month.
```

**Day Class**

```java
package Calendar;

import java.util.*;

class Day {
    private ArrayList appointments;
    private int dayNumber;
    private String name;

    public Day(String theName, int theNumber) {
        name = theName;
        dayNumber = theNumber;
        appointments = new ArrayList();
    }

    public void addAppointment(String description) {
        appointments.add(description);
    }

    public int getDayNumber() {
        return dayNumber;
    }

    public String getName() {
        return name;
    }
    ... }
```
class Day {

// constructor, addAppointment, and accessors omitted.
public boolean anyAppointments() {
    if (appointments.size() == 0) { return true; }
    else { return false; } }

public String toString() {
    String result="(" + dayNumber + ")
";
    String[] allAppointments = new String[appointments.size()];
    appointments.toArray(allAppointments);
    for (int i=0; i<allAppointments.length; i++){
        result += "  " + allAppointments[i] + "\n"; }
    return result; } }

MonthlyCalendar Class

public class MonthlyCalendar {

// Some details omitted ...
private int numberOfDays;
    private int startingDayNum;
    private String monthName;
    private Day[ ] calendar;

    public MonthlyCalendar(String theMonthName, int theNumberOfDays, int theStartingDay) {

        monthName = theMonthName;
        numberOfDays = theNumberOfDays;
        startingDayNum = theStartingDay;
        calendar = new Day[NUMBER_OF_WEEKS];
        int currentDayNumber = 1;
        int dayIndex = 0;
        for (int week=0; week<NUMBER_OF_WEEKS; week++) {
            calendar[week] = new Day[DAYS_PER_WEEK];
            // Create the Day objects for the current week
            for (int dayOfWeek=0; dayOfWeek<DAYS_PER_WEEK; dayOfWeek++) {
                if (dayIndex >= (startingDayNum - 1)) {
                    calendar[week][dayOfWeek] =
                    new Day(mapIntToDayName(dayOfWeek), currentDayNumber);
                    currentDayNumber++; }
            }
        }
    } 
} }... } // More to come.
**MonthlyCalendar Class (continued)**

Three Support functions:

public String mapIntToDayName(int value) {
    String[] dayNames = {
        "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};
    return dayNames[value];
}

public void addAppointment(int theDayNumber, String description) {
    for (int weekNum=0; weekNum<calendar.length; weekNum++) {
        for (int dayNum=0; dayNum<calendar[weekNum].length; dayNum++) {
            Day day = calendar[weekNum][dayNum];
            if ((day!=null) && (day.getDayNumber() == theDayNumber)) {
                day.addAppointment(description); }
        }
    }
}

public String toString() {
    String result = monthName + "\n";
    for (int weekNum=0; weekNum<calendar.length; weekNum++) {
        result += "Week[" + (weekNum+1) + "]:\n";
        for (int dayNum=0; dayNum<calendar[weekNum].length; dayNum++) {
            Day day = calendar[weekNum][dayNum];
            if ((day!=null) && (day.anyAppointments())) {
                result += " " + calendar[weekNum][dayNum]; }
        }
    }
    return result;
}

**Example Output**

```java
MonthlyCalendar monthlyCalendar = new MonthlyCalendar("May", 31, 6);

monthlyCalendar.addAppointment(2, "Clean Car");
monthlyCalendar.addAppointment(2, "Pay Mortgage");
monthlyCalendar.addAppointment(3, "Parent Meeting");
monthlyCalendar.addAppointment(28, "Submit project");

System.out.println(monthlyCalendar);
```