Arrays in Java – Arrays of Objects

Elements of an array can be of any type, and when applied to objects, this can prove very practical and useful. For example, consider collecting data on a group of children. We might have multiple arrays to store their names, ages, heights, and genders.

```
String[] name = new String[MAX_CHILDREN];
int[] age = new int[MAX_CHILDREN];
double[] height = new double[MAX_CHILDREN];
char[] gender = new char[MAX_CHILDREN];
```

Arrays of the same length, carrying data on different aspects of a problem, are sometimes called *parallel arrays*. Rather than four parallel arrays for our study, we could use a single array of objects.

```
class Child
{
  private String name;     // family, given
  private int age;         // in years
  private double height;   // in metres
  private char sex;        // M or F
}

Child[] patient = new Child[MAX_CHILDREN];
```

In this case, all of the information about the first child can be referenced through `patient[0]`, such as `patient[0].age` or `patient[0].gender`.

There are a number of advantages to using an array of objects rather than multiple arrays of data.

1. A method to process data about one object can be given a single parameter rather than many parameters.

2. If the program is later modified to include different kinds of data, we only need to change the fields in the object class. The parameter lists of any methods remain unchanged.

3. By making the class fields private, we can ensure the data will be handled appropriately through the use of accessor and mutator methods.

Exercises

1. For the child development study discussed above:

   (a) write a method that could be added to the `Child` class that prints the data (name, age, height, and gender) for a child on two lines with the name on the first line and the other data on the second line. Print the height in centimetres, rounded to the nearest centimetre.

   (b) write a fragment that uses this method to print the data on all the children.

   (c) write a method that would produce the same output at the method from part (a) if we had chosen to use parallel arrays rather than an array of objects to represent our data.