Ejercicios Arrays

Programación II

Resuelva los siguientes ejercicios extraídos del capítulo 12 del libro ThinkJava:

43) Write a method called cloneArray that takes an array of integers as a parameter, creates a new array that is the same size, copies the elements from the first array into the new one, and then returns a reference to the new array.

44) Write a method called randomDouble that takes two dou- bles, low and high, and that returns a random double x so that low ≤ x < high.

45) Write a method called randomInt that takes two arguments, low and high, and that returns a random integer between low and high, not including high.

46) Encapsulate the code in Section 12.10 in a method called makeHist that takes an array of scores and returns a histogram of the values in the array.

47) Write a method named areFactors that takes an integer n and an array of integers, and that returns true if the numbers in the array are all factors of n (which is to say that n is divisible by all of them). HINT: See Exercise 6.1.

48) Write a method that takes an array of integers and an integer named target as arguments, and that returns the first index where target appears in the array, if it does, and -1 otherwise.

49) Some programmers disagree with the general rule that variables and methods should be given meaningful names. Instead, they think variables and methods should be named after fruit.
For each of the following methods, write one sentence that describes abstractly what the method does. For each variable, identify the role it plays.

```
public static int banana(int[] a) {
   int grape = 0;
   int i = 0;
   while (i < a.length) {
      grape = grape + a[i];
i++; }
   return grape;
}

public static int apple(int[] a, int p) {
   int i = 0;
   int pear = 0;
   while (i < a.length) {
      if (a[i] == p) pear++;
i++; }
   return pear;
```

```
        }

ublic static int grapefruit(int[] a, int p) {
   for (int i = 0; i<a.length; i++) {
      if (a[i] == p) return i;
   }

return -1;

}
```

The purpose of this exercise is to practice reading code and recognizing the computation patterns we have seen.