Chapter 7 Array

An array is a collection of items of the same type;

Creating and accessing arrays

Double[] temperature = new double[7];

The integer expression within the square brackets is called an index or a subscript; Note that the numbering starts with 0, not 1;

[]: square brackets;

For( int index = 0; index < 7; index ++ )

Temperature[index] = keyboard.nextDouble();

The user could type the seven values on separate lines or all on one line, separated by spaces.

Base\_Type[] Array\_Name = new Base\_Type[Length];

Int[] pressure = new int[100];

Int[] pressure;

Pressure = new int[100];

Base\_Type[] Array\_Name = new Base\_Type[Length];

E.g:

char[] symbol = new char[80];

double[] reading = new double[100];

Species[] specimen = new Species[80];

How to use square brackets with arrays:

There are three different ways to use square brackets [] with an array name. They can be used:

With a data type when declaring an array.

int[] pressure; declares—but does not allocate memory for –pressure as an array of integers;

To enclose an integer expression when creating a new array. For example :

Pressure = new int[100]; allocates memory for the array pressure of 100 integers.

To name an indexed variable of the array . For example, pressure[3] in the following two lines is an indexed vartiable:

Pressure[3] = keyboard.nextInt();

System.out.println(“You entered” + pressure[3]);

The instance variable length

An array is a kind of object, and like other objects, it might have instance variables.

Array.length

Your program cannot assign a value to the instance variable length, as it is a final variable;

An array can be initialized at the time that it is declared;

Arrays can be used as instance variables in classes. Methods can have an indexed variable or an entire array as an argument and can return any array; In short , arrays can be used with classes and methods just as other objects can.

Indexed variables as method arguments:

An indexed variable can be an argument to a method in exactly the same way that any other variable of the array’s base type can be an argument;

If the base type of the array is primitive type, the method cannot change the value of a[i];

On the other hand , if the base type of the array is a class type,the method receives a reference to a[i] and the method can change the state of the object named by a[i] ;

A parameter can represent an entire array ;

Public class SampleClass

{

Public static void incrementArrayBy2(double[] anArray)

{

For( int i = 0; i < anArray.length; i++)

anArray[i] = anArray[i] +2;

}

}

A method can change the values of the elements in its array argument;

When you specify an array parameter, you give the base type of the array, but you do not fix the length of the array;

Characteristics of Array Arguments:

No square brackets are written when you pass an entire array as an argument to a method;

An array of any length can be the argument corresponding to an array parameter;

A method can change the values in an array argument;

Arguments for the method main:

Public static void main(String[] args)

Args is an array whose base type is String. The method main takes an array of String values as an argument.

When you run your program, main is invoked automatically and given a default array of strings as a default argument. And you can if you like , provide additional strings when you run a program, and those strings will automatically be made elements of the array args that is provided to main as an argument.

Java TestProgram Sally Smith;

This command sets args[0] to “ Sally “ and args[1] to “Smith”

You can pass an array of strings to main as an argument;

Arrays are objects and so the assignment operator = and the equality operator == behave in the same way with arrays as they do with the kinds of objects we saw before discussing arrays;

The entire array contents ; the contents of all of the indexed variables are stored together in one , possibly large, section of memory. In this way , the location of the entire array contents can be specified by one memory address;

A variable for an object really contains the memory address of the object;

Int[] a = new int[3];

Int[] b = new int[3];

b = a ;

The assignment gives the array variable b the same memory address as the array variable a ;

And the equality operator == tests whether two arrays are stored in the same place in the computer’s memory;

Arrays do not belong to any class. Some other features of class objects such as inheritance do not apply to arrays;

Java method may return an array. To have it do so, you specify the method’s return type in the same way that you specify the type of an array parameter.

A method can return an entire array;

Returning an Array

Public static Base\_Type[] Method\_name(parameter\_list)

{

Baser\_Type[] temp = new Base\_Type[Array\_Size];

Statements\_To\_Fill\_Array

Return temp;

}

7.3 Programming with arrays and classes

One way to use an array for a special purpose is to make the array an instance variable of a class; The array is accessed only through the class methods; The array is accessed only through the class methods;

An array of strings as an instance variable ; Access to the strings in a list is only via methods;

Partially filled Array :

Distinguish between the number of entries that you use in an array and the array’s capacity. The capacity of an array a is a.length; This is the number of array elements that are created for you when you define the array. You might not use all of these elements. In this case , you need to keep track of how many you have used. Typically you will have two integers: one is the capacity of the array and one is the number of elements you have used.

Base\_Type[]…[] Array\_Name = new Base\_Type[Length\_1]…[Length\_n];

Multidimensional-Array parameters and returned values:

Methods can have parameters and return values that are multidimensional arrays.

Return a multidimensional array:

Public static Return\_Type Method\_Name(Base\_Type[]…[] Parameter\_Name)