Universidad Andrés Bello®
Conectar · Innovar · Liderar

FUNDAMENTOS DE PROGRAMACIÓN EN JAVA

PROGRAMACIÓN BÁSICA EN JAVA

El entorno Java para la Programación

Arreglos y colecciones

Arreglos

Hace algunas clases atrás se definió el concepto de arreglo. Se mencionó que era una zonade memoria, que puede almacenar un conjunto de N datos del mismo tipo. Un array en Java puede considerarse un "objeto especial", o un tipo de dato diferente, ya que se creacon la sentencia new, como el resto de los objetos de una clase (este tema lo veremos en detalle en clases siguientes). Sin embargo, no hay una clase específica en Java que definaa los arreglos o arrays como un todo.

La declaración de un arreglo es similar a la declaración de una variable, la sintaxis es lasiguiente:

```
tipoDato nombreArreglo[ ];
```

Algunos ejemplos de declaraciones de arreglos serían:

```
int edad[ ]; //Declara arreglo de enteros llamado edad.

float nota[ ]; //Declara arreglo de reales llamado nota.

String nombre[ ]; //Declara arreglo de caracteres llamado
nombre.
```

Después de declarar un arreglo, este debe inicializarse; para ello se usa la palabrareservada new y se define la cantidad de posiciones que el arreglo puede contener parael almacenamiento de valores. La sintaxis a utilizar es la siguiente:

```
Arreglo = new TipoDato[tamaño];
```

A continuación se indican algunos ejemplos de asignación de tamaño a un arreglo.

```
edad = new int [10]; //Creación del arreglo edad con 10 posiciones.

nota = new float [15]; //Creación del arreglo nota con 15 posiciones.

nombre = new String [5]; //Creación del arreglo nombre con 5 posiciones.
```

Antes de continuar, es importante detenerse y considerar la siguiente sentencia:

```
int [ ] donacionesPorMes = new int [12];
```

A través del ejemplo anterior, se puede identificar errores habituales que cometen algunos desarrolladores. Por lo mismo, siempre es bueno tener en consideración las siguientes afirmaciones, más comunes de lo que se espera:

1. Los índices del arreglo parten en 0 y terminan en 12.
   - Falso, van de 0 a 11.
2. El número de elementos total es 11.
   - Falso, son 12 elementos en total.
3. Para acceder al último elemento, se debe usar la expresión donacionesPorMes[12].
   - Falso, el índice 12 no existe, y el uso de esa expresión daría lugar a un erroren tiempo de ejecución del tipo “ArrayIndexOutOfBoundsException”.
4. La primera posición de un arreglo en Java es la posición 0 y la última es 12
   - Falso, si n es el tamaño de un arreglo, el último índice será n-1. En el casoejemplo es 12-1 = 11.

En Java también es posible declarar e inicializar un objeto en la misma línea de código. A continuación se indican ejemplos para lograr este comportamiento.

```
int edad [ ] = new int [10]; //Declaración y creación de
un arreglo llamado edad de 10 posiciones de tipo entero.

float nota [ ] = new float [15]; //Declaración y creación
de un arreglo llamado nota de 15 posiciones de tipo float.

String nombre [ ] = new String [5]; //Declaración y
creación de un arreglo llamado nombre de 5 posiciones de
tipo String.
```

Ejemplo 1: Desarrolle un programa en lenguaje Java que muestre la declaración, creación, inicialización y consulta de un arreglo llamado edadcon 3 posiciones de tipo entero.

```
public class Ejemplo1{
    public static void main(String[] args){
        int edad[]= new int[3];

        edad[0]=18;
        edad[1]=20;
        edad[2]=15;

        for(int i=0; i<3; i++){
            System.out.println("El valor del arreglo edad
en la posicion "+i+" es "+edad[i]);
        }
    }
}
```

También es posible inicializar un arreglo en la misma línea donde es declarado y creado.

Ejemplo 2: Desarrolle un programa que cree e inicialice en la misma línea de código un arreglo de 3 posiciones de tipo carácter con los nombres de 3 personas diferentes; posteriormente se debe imprimir el contenido del arreglo.

```
public class Ejemplo2 {
    public static void main(String[] args){
        String [] nombre = new String[]{"Juan", "Pedro",
"Diana"} ;
        for (int i = 0; i < 3; i++){
            System.out.println(nombre[i]);
        }
    }
}
```

Si se desea saber el largo del arreglo, basta con usar la sintaxis nombreDelArray.length, la cual devuelve un valor entero (int) con el número de elementos que forman el array. Siguiendo la misma idea del ejemplo anterior, quedaría de la siguiente manera:

```
System.out.println ("El número de elementos en el array
nombre es de " + nombre.length );
```

Es posible usar ciclos del tipo for each con arreglos, lo cual tiene como beneficio quenunca saldrá de los límites del arreglo, y que no es necesario conocer el tamaño del arreglo.

Ejemplo 3: Desarrolle un programa que cree un arreglo de cadenas de texto de largo 10, y despliegue el contenido de sus celdas por medio de un ciclo for each.

```
public class Ejemplo3 {

     public static void main(String[] args) {
          String [ ] misNombres = new String [10];

          misNombres[0] = "Juan";
          misNombres[1] = "Pedro";
          misNombres[2] = "Diego";
          //Se asignan los demás valores del arreglo

          for (String tmpObjeto : misNombres) {
               System.out.println (tmpObjeto);
          }
     }

}
```

Colecciones

Los arreglos son una excelente herramienta de programación, principalmente cuando la cantidad de posiciones del mismo se mantiene constante. Sin embargo, al momento de incluir más elementos dentro de un arreglo, o al momento de querer eliminar posiciones del mismo, los arreglos son ineficientes. En el lenguaje de programación Java existen otras estructuras de almacenamiento conocidas como colecciones. Una colección permitealmacenar información en memoria de manera dinámica, es decir, permite el aumento o disminución del tamaño de la colección de acuerdo con la necesidad de la aplicación.

En la siguiente tabla se muestran los distintos tipos de elementos de Java que cumplen esta condición.

Elemento	Descripción
Collection	Define un conjunto de métodos comunes a las demás interfaces o clases que se derivan de la interfaz Collection.
List	Maneja de manera ordenada los elementos introducidos de acuerdo con el orden de inserción. Permite elementos duplicados.
Set	Colección desordenada de objetos. No permite elementos duplicados.
ArrayList	Clase concreta que implementa la interfaz List. Esta clase se basa en un arreglo dinámico. Permite un acceso rápido a los elementos de la colección pero es lenta para la inserción o eliminación de objetos.
LinkedList	Clase concreta que implementa la interfaz List, esta clase se basa en una lista enlazada. Es muy útil cuando la cantidad de elementos de la colección es muy variable. El acceso a un elemento particular de la colección es más lento
HashSet	Clase concreta que implementa la interfaz Set. Los elementos de esta clase no permiten duplicados y no se almacenan en orden no determinado
TreeSet	Clase concreta que implementa la interfaz Set. Los elementos de esta clase no permiten duplicados y sus elementos se ordenan de manera ascendente

Uso de la clase ArrayList

La clase ArrayList en Java es una clase que permite almacenar datos en memoria de formasimilar a los Arrays pero con la ventaja de que el número de elementos que almacena, lohace de forma

dinámica, es decir, que no es necesario declarar su tamaño como pasa conlos Arrays.

Ejemplo 4: Defina un elemento de tipo ArrayList, y agregue a éste 3 elementos. Una vez hecho lo anterior, despliegue todos sus valores y, de forma separada, el último valor.

```
public class Ejemplo4 {
     public static void main(String arg[]) {

          ArrayList<Integer> listaNums = new
ArrayList<Integer>();
          listaNums.add(8);
          listaNums.add(3);
          listaNums.add(5);
          System.out.println("Lista de números: " +
listaNums);
          System.out.println("Número posicion 2: " +
listaNums.get(2));

          //Asigna un valor en la primera posición, y
despliega nuevamente los valores
          listaNums.set(0, 15);
          System.out.println("Lista de números:" +
listaNums);
     }
}
```

Anexo: Referencias

1.- Arreglos en Java
Referencia: http://www.manualweb.net/java/arrays-java/

2.- Ejercicios de ejemplo con arreglos en Java
Referencia: http://puntocomnoesunlenguaje.blogspot.com/2012/07/arreglos-en-java-calcular-media.html