CENTRO DE TECNOLOGIAS DE INFORMACION Y COMUNICACION

JAVA BASICO

Instructor: LENIN LEON
Email: leninleon0720@gmail.com
Celular: 6234-5608

Arreglos

Se pueden declarar arreglos de cualquier tipo de dato:
char letras[];
Point punto[];

Arreglos

En Java, un arreglo es un objeto, aun cuando el arreglo es de tipos de datos primitivos, y como con los demás objetos, la declaración no crea el objeto en sí. De modo que estas declaraciones no crean los arreglos, solo hacen referencia a variables que pueden ser usadas para accesar al arreglo. También se pueden crear arreglos con la siguiente sintaxis:
char [] letras;

Arreglos

Ambas declaraciones son válidas, pero la segunda tiene la ventaja de que, si se declaran más arreglos, sólo basta con poner el nombre de cada arreglo:

char letras[], numeros[];
por:
char [] letras, numeros;

Arreglos
Creación e inicialización

Para crear los arreglos se usa la palabra new:

```
letras = new char[20];
punto = new Point[100];
```

La primera línea crea un arreglo de 20 valores char. La segunda línea crea un arreglo de 100 variables de tipo Point, que es un objeto. Sin embargo, no crea los 100 objetos Point. Se tiene que crear separadamente cada objeto:

```
punto[0] = new Point();
punto[1] = new Point();
punto[2] = new Point();
...
```

Arreglos
Creación e inicialización

Cuando se crea un arreglo, cada elemento es inicializado. En el caso del arreglo letras de tipo char, cada valor es inicializado al carácter nulo (\u0000). En el caso del arreglo punto, cada valor fue inicializado a null, indicando que no hay referencia al objeto Point. Después de la asignación punto[0] = new Point(), el primer elemento del arreglo se refiere a un objeto Point. Java permite una manera fácil de crear arreglos con valores iniciales:

String nombres[] = {"Juan","Pedro","Luis"};

Arreglos
Creación e inicialización

La línea anterior es similar a:

```
String nombres[];
nombres = new String[3];
nombres[0] = "Juan";
nombres[1] = "Pedro";
nombres[2] = "Luis";
```

Esta forma se puede aplicar tanto a tipos de datos primitivos como a objetos, por ejemplo:

```
Color paleta[] = {Color.red,Color.green,Color.blue};
```

Arreglos
Creación e inicialización

Java no provee arreglos multidimensionales, en cambio, como se pueden crear arreglos de cualquier tipo, se crean arreglos de arreglos de arreglos...

```
int dosDim[][] = new int[4][];
dosDim[0] = new int[5];
dosDim[1] = new int[5];
```

Arreglos
Creación e inicialización

Primero se declara un arreglo de arreglos de int. Luego se crea el arreglo con cuatro arreglos de int.
Finalmente, se crea cada arreglo con cinco valores int.

Arreglos
Creación e inicialización

Debido a que no hay arreglos multidimensionales, se pueden crear arreglos de arreglos no-rectangulares, por ejemplo:

```
dosDim[0] = new int[2];
dosDim[1] = new int[5];
dosDim[2] = new int[8];
```

Arreglos
Creación e inicialización

Pero esta forma de arreglos no es muy común y es tediosa de programar. La forma rectangular de los arreglos es la más común, por lo que Java provee una forma fácil de crear arreglos bidimensionales:

```
int dosDim[][] = new int[4][5];
```

esto crea un arreglo de cuatro arreglos de cinco enteros cada uno.

Control del tamaño del arreglo

En Java todos los índices de los arreglos empiezan en cero. El número de elementos en un arreglo es almacenado como parte del objeto arreglo. El valor es usado para realizar evaluaciones de límite en todos los accesos en tiempo de ejecución. Si hay un acceso fuera del límite del arreglo, se crea una excepción.

Control del tamaño del arreglo

El límite del ciclo se determina por la comparación con lista.length, en vez de comparar directamente contra el valor 10. Esto es más robusto cuando se trata de dar mantenimiento al programa.

Copiado de arreglos

Una vez creado, un arreglo no puede cambiar de tamaño. Sin embargo, se puede usar la misma variable de referencia para un arreglo nuevo:

```
int elementos[] = new int[6];
elementos = new int[10];
```

Copiado de arreglos

De esta manera, se pierden los seis valores del arreglo elementos, a menos que se hayan almacenado en otro lugar. Java provee un método en la clase System para copiar arreglos. Este método es arraycopy(). Por ejemplo:

```
// arreglo original
int elementos[] = {1,2,3,4,5,6};
// arreglo destino
int masnumeros[] = {10,9,8,7,6,5,4,3,2,1};
// copiar todos los valores de elementos en masnumeros
System.arraycopy(elements,0,masnumeros,0,elements.length)
;
```

En este punto, el arreglo más números tiene los siguientes valores: 1,2,3,4,5,6,4,3,2,1.

Laboratorio 3

En España cada persona está identificada con un Documento Nacional de Identidad (DNI) en el que figura un número y una letra, por ejemplo 56999545W

La letra que sigue al número se calcula siguiendo la metodología que vamos a indicar. Crea un programa que calcule la letra de un DNI a partir del número de DNI que introduzca el usuario. Es decir, se debe pedir el DNI sin la letra por teclado y el programa nos devolverá el DNI completo (con la letra).

Para calcular la letra, se debe tomar el resto de dividir nuestro número de DNI entre 23. El resultado debe estar por tanto entre 0 y 22.

Laboratorio 3

Crea un método obtenerLetra(int numeroDNI) donde según el resultado de la anterior fórmula busque en un array de caracteres la posición que corresponda a la letra. Esta es la tabla de caracteres:
Posición 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

Letra T R W A G M Y F P D X B N J Z S Q V H L C K E
Por ejemplo, si introducimos el DNI 20267079, el resto de dividirlo por 23 sería 8, luego la letra sería la P, que es la que ocupa esa posicion en la matriz de caracteres.