Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II
Practica de Laboratorio 1:
Arreglos Estándar

I

OBJETIVOS

- Crear e inicializar arreglos
- Uso del atributo length de arreglos
- Copiar valores de un arreglo a otro mediante System.arraycopy()
- Crear métodos que reciban arreglos como parámetros
- Valorar el uso de arreglos estándar frente a las variables simples

II

MARCO TEORICO

1. Creación e Inicialización de Arreglos

Un arreglo representa una colección de elementos del mismo tipo bajo un mismo nombre. En la Figura 1 se muestra el arreglo phoneList que contiene 5 elementos.

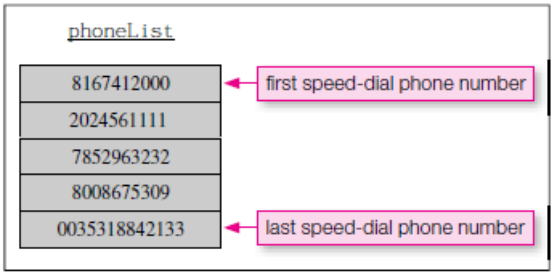

Figura 1: Ejemplo de arreglo de 5 elementos

Para acceder a los elementos de un arreglo se hace uso de los índices. El primer índice siempre es el 0. En la figura 2 se muestra como se accede a los elementos del arreglo

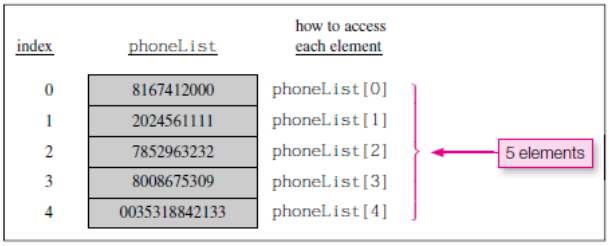

Figura 2: Ejemplo de acceso a elementos mediante el índice

2. Creación e Inicialización de Arreglos

Estilo 1:

```
double[] temperatures = new double[5];
for (int i=0; i<5; i++)
{
  temperatures[i] = 98.6;
}
```

declare and create array

assign a value to the i^{th} array element

Estilo 2:

```
double[] temperatures = {98.6, 98.6, 98.6, 98.6, 98.6};
```

3. Valores por defecto

Array Element's Type	Default Value
integer	0
floating point	0.0
boolean	false
reference	null

4. Propiedad length de los arreglos

Suponga que creamos e inicializamos un arreglo con 5 elementos:

```
String[] colors = {"blue", "gray", "lime", "teal", "yellow"};
```

Para imprimir el contenido tenemos dos formas de hacerlo

Versión 1: utilizando un número

```
for (int i=0; i<5; i++)
{
  System.out.println(colors[i]);
}
```

hard-coded array size

Versión 2: utilizando length

```
for (int i=0; i<colors.length; i++)
{
  System.out.println(colors[i]);
}
```

number of elements in the array

5. Copia de Arreglos

- Podríamos crear nuestros propios métodos que copien elemento a elemento
- Pero también podemos usar `System.arraycopy`
- Permite copiar un número de elementos desde cualquier lugar en un arreglo a cualquier lugar en otro arreglo

```
System.arraycopy(preciosEnero,0,preciosFebrero,0,6);
preciosFebrero[1]=10.99;
```

- Argumentos:
  - Nombre arreglo origen
  - Índice del primer elemento del origen a copiar
  - Nombre del arreglo de destino
  - Índice del primer elemento a reemplazar en el destino
  - Número total de elementos a copiar

6. Comparación de Arreglos

- Se podría pensar en usar a.equals(b), pero eso sólo comparará las referencias
- Importar java.util.*
- Usamos métodos booleanos de la clase `Arrays`
- `Arrays.equals()` para unidimensionales y `Arrays.deepEquals()` para cualquier dimensión
- `boolean Arrays.deepEquals(Object[] a1, Object[] a2)`

II

CONSIDERACIONES DE EVALUACIÓN

- No deberá utilizar constructores no vistos en clase
- No podrá modificar el código base entregado para este laboratorio
- Deberá utilizar nombre de variables significativos
- Deberá realizar pruebas adicionales
- El alumno deberá indicar en su código con quien colaboró
- El alumno será requerido de realizar modificaciones en su código y responder a preguntas sobre el mismo
- Todos los ejercicios deberán traerse terminados en caso de ser tarea para la casa
- Si tiene ejercicios sin terminar no importa, se revisará el avance y se discutirá sobre las dificultades encontradas.

III

POLITICA DE COLABORACION

La política del curso es simple, a menos que se exprese lo contrario en el laboratorio, siéntase libre de colaborar con sus compañeros en todos los laboratorios, pero debe notificar expresamente con quien ha colaborado. La colaboración con alumnos, que no están matriculados en el curso está prohibida. Los laboratorios y asignaciones han sido desarrollados para

ayudarlo a comprender el material. Conozca su código y esté preparado para revisiones individuales de código. Durante las revisiones es probable que se le pida realizar modificaciones y justificar sus decisiones de programación. Cada uno de sus ejercicios debe iniciar de la siguiente forma

```
// Laboratorio  Nro x - Ejerciciox
// Autor: mi nombre
// Colaboró : el nombre
// Tiempo :
```

IV

INDICACIONES GENERALES

a. Todos los ejercicios deberán ser guardados en el mismo Proyecto
b. El Proyecto deberá tener el nombre del Laboratorio y el nombre del alumno, así por ejemplo:
   Laboratorio 1 – Juan Perez
c. Cada Clase deberá tener el nombre del ejercicio, así por ejemplo:
   Ejercicio1
d. Utilice nombres de variables significativos y todas las recomendaciones de estilo
e. Especialmente, su código deberá estar correctamente indentado
f. Deberá pasar TODOS los casos de prueba

V

ACTIVIDADES

Enunciado: antes de simular una batalla entre dos ejércitos, debemos considerar que cada ejército está compuesto por soldados. Dada su experiencia con videojuegos de estrategia, ¿qué datos de los soldados son importantes? (considerar que cada soldado tendrá que ser identificado individualmente). Usando lluvia de ideas, los estudiantes indican que necesitamos conocer su nombre, nivel de vida, velocidad, etc.

Actividad 1: escribir un programa donde se creen 5 soldados considerando sólo su nombre. Ingresar sus datos y después mostrarlos.

Restricción: se realizará considerando sólo los conocimientos que se tienen de FP1 y sin utilizar arreglos estándar, sólo usar variables simples.

Actividad 2: escribir un programa donde se creen 5 soldados considerando su nombre y nivel de vida (aleatorio entre 1 y 5). Ingresar sus datos y después mostrarlos.

Restricción: se realizará considerando sólo los conocimientos que se tienen de FP1 y sin utilizar arreglos estándar, sólo usar variables simples.

Actividad 3: escribir un programa donde se creen 5 soldados considerando sólo su nombre. Ingresar sus datos y después mostrarlos.
Restricción: aplicar arreglos estándar.

Actividad 4: escribir un programa donde se creen 5 soldados considerando su nombre y nivel de vida. Ingresar sus datos y después mostrarlos.
Restricción: aplicar arreglos estándar. (Todavía no aplicar arreglo de objetos)

Actividad 5: escribir un programa donde se creen 2 ejércitos, cada uno con un número aleatorio de soldados entre 1 y 5, considerando sólo su nombre. Sus datos se inicializan automáticamente con nombres tales como “Soldado0”, “Soldado1”, etc. Luego de crear los 2 ejércitos se deben mostrar los datos de todos los soldados de ambos ejércitos e indicar qué ejército fue el ganador.
Restricción: aplicar arreglos estándar y métodos para inicializar los ejércitos, mostrar ejército y mostrar ejército ganador. La métrica a aplicar para indicar el ganador es el mayor número de soldados de cada ejército, puede haber empates. (Todavía no aplicar arreglo de objetos)

Entregable: videojuego v1.0