Departamento de Ciencias de la Computación
UNIVERSIDAD DE CHILE

Auxiliar 8
Arreglos y Matrices

Computación I – CC1001-05

Arreglo

- Un arreglo es una estructura de datos que permite almacenar “dentro” de una misma variable una serie de datos del mismo tipo.
- Los arreglos deben ser declarados y es necesario fijarles un tamaño.

```
int[] a = new int[10]; // Arreglo de enteros de tamaño 10
double[] b = new double[100]; // Arreglo de reales de tamaño
   100
int[][] m = new int[2][2]; // Matriz cuadrada de 2x2
String[] s = new String[5]; // Arreglo de Strings de tamaño 5
```

Arreglo

- Los arreglos además poseen algunas propiedades interesantes
  - Se puede conocer su tamaño aplicándoles “length” (sin paréntesis

```
int[] a = new int[10];
U.println(“a tiene “ + a.length + “ elementos”);
```

  - El primer elemento es el 0 y el último es el length-1

```
for (int i=0; i<n; i++)
   a[i] = U.azar(1, 100);
```

Ejercicio

- Escriba un programa que lea un archivo de notas.txt (con 100 registros) y que luego haga una tabla en pantalla con las frecuencias de notas:

```
Intervalo     Frecuencia
6.0 – 7.0          xxx
5.0 – 5.9          xxx
4.0 – 4.9          xxx
3.0 – 3.9          xxx
2.0 – 2.9          xxx
1.0 – 1.9          xxx
```

Ejercicio

Escribamos el programa como si existiera una función que calculara la frecuencia de valores dentro del arreglo en un intervalo fijo. Esto ya lo sabemos hacer, ya que lo único diferente es pasar el arreglo a un método por parámetro:

```
import java.io.*;
public class Distribucion {
   static public void main (String[] args) throws Exception {
      double[] notas = double[100];
      BufferedReader fd = new BufferedReader(new
  FileReader("notas.txt"));
      String linea; int n=0;
      while((linea = fd.readLine()) != null)
         notas[n] = Double.parseDouble(linea);
      fd.close();
      U.println("Intervalos \t Frecuencia");
      U.println("6.0 – 7.0 \t " + frecuencia(notas, 6.0, 7.0));
      ...
      U.println("1.0 – 1.9 \t " + frecuencia(notas, 1.0, 1.9));
   }
```

Ejercicio

Ahora implementaremos el método “frecuencia” que se preocupará de calcular la frecuencia de valores del arreglo dentro de un intervalo fijo.

```
    public int frecuencia(double[] notas, double min, double max) {
       int cont = 0;
       for(int i=0; i<notas.length; i++)
          if (notas[i] >= min && notas[i] <= max) cont++;
       return cont;
    }
 }
```

Ejercicio

- Escriba un programa solicite el tamaño de una matriz (2 dimensiones) y que genere al azar un laberinto en pantalla. Las condiciones para generar el laberinto son:
  - El contenido de una celda es un número 0, 1 o 2.
    - 0 es pasillo
    - 1 es muro
    - 2 es trampa

  Al final deben imprimir en pantalla el laberinto poniendo una “X” cuando es muro y un blanco “ “ cuando no (las trampas no se muestran)

Ejercicio


```
public class Laberinto {
   static public void main (String[] args) throws Exception {
```

Solicitamos primero las dimensiones del laberinto en cantidad de FILAS y COLUMNAS.

```
int m = U.readInt("   Filas: ");
int n = U.readInt("Columnas: ");
```

Creamos el laberinto con el tamaño indicado y luego vamos poniendo al azar si es una trampa, muro o pasillo.

```
int[][] lab = new int[m][n];
for (int i=0; i<m; i++)
   for (int j=0; j<n; j++)
      lab[i][j] = U.azar(0, 2);
```

Ejercicio

Lo que nos queda es solo mostrar el contenido del laberinto en pantalla.

```
        for (int i=0; i<m; i++) {
           for (int j=0; j<n; j++) {
              if (lab[i][j] == 1)
                 U.print("X");
              else
                 U.print(" ");
           }
           U.println("");
        }
    }
}
```

Nótese que lo que aparece en pantalla es posible que no sea ni lógicamente un laberinto, porque no tiene lógica de pasillos, etc. Sin embargo esa puede ser una optimización.

Ejercicio con Nota

- Se desea hacer un “generador de manos” de cartas, que permita simular el repartir 4 series de “manos” compuestas por 13 cartas de un naipe inglés. Para ello, se le pide generar un programa que seleccione al azar los números y las pintas en un arreglo de Strings, en donde cada posición es un String del tipo: 1P, 5C, 3D, JT (en donde P = Pica, C = Corazón, D = Diamante y T = Trébol). El programa luego de generar cada mano aleatoriamente, debe mostrarlas en pantalla como:

```
Mano 1 = QT ... 3D
Mano 2 = 3P ... QD
Mano 3 = 2P ... JT
Mano 4 = 4T ... 6P
```

  Recuerde que cada mano no debe repetir ninguna carta.