

Programación estructurada

Diseño estructurado

Verónica E. Arriola-Rios

Facultad de Ciencias, UNAM

14 de septiembre de 2022



Paradigma

1 Paradigma

2 Diseño

3 Java

4 Bibliografía

Programación imperativa

- En la *programación imperativa* se dan órdenes a la computadora sobre lo que debe hacer.
- La *programación estructurada* es un tipo de programación imperativa, caracterizada por el uso de estructuras de control y la organización de las instrucciones en funciones.

Los elementos de un programa. López Román 2011

Estructuras de datos. Son las formas de representación interna de datos de la computadora, es decir, son los tipos de datos que maneja la máquina.

Operaciones primitivas. Son las acciones básicas que la computadora “sabe” hacer, y que se ejecutan sobre los datos.

Estructuras de control. Son las formas lógicas de funcionamiento de la computadora mediante las que se dirige el orden en que deben ejecutarse las instrucciones del programa.

Temas

1 Paradigma

- Estructuras de datos
- Operaciones primitivas
- Estructuras de control

Variables

- La información se accede a través de variables.
- Para poder almacenar información se deben apartar localidades de memoria *declarando* variables.
- Una declaración especifica el *tipo de dato* a manipular y el *nombre* con el cual se referirá a la localidad de memoria.

```
1 double x;  
2 boolean valor;
```

- En la *definición* se asigna un valor a la variable, quedando éste almacenado en esa localidad de memoria.

```
1 x = 3.1416;  
2 valor = true;
```

Temas

1 Paradigma

- Estructuras de datos
- Operaciones primitivas
- Estructuras de control

Operaciones primitivas

Ejemplos de operaciones primitivas son:

- Declarar variables.
- Asignar valores.
- Operaciones aritméticas fundamentales:
 - suma +
 - resta -
 - multiplicación *
 - división /

Temas

1 Paradigma

- Estructuras de datos
- Operaciones primitivas
- Estructuras de control

Estructuras de control

Definición

Las *estructuras de control* son las formas lógicas como puede comportarse una computadora y mediante las cuales se dirige el funcionamiento de la misma. Se le da orden lógico a las operaciones que actúan sobre los datos. López Román 2011

Expondremos aquí, 4 formas de organizar dichas operaciones:

- 1 Secuenciación
- 2 Condicional
- 3 Repetición
- 4 *Recursión

Organización de los enunciados

- *Secuenciación*. Sucesión ordenada de instrucciones.
- *Iteración*. Repite un bloque de enunciados secuenciales.
- *Ejecución condicional*. Elige uno o más bloques de enunciados secuenciales, dependiendo de condiciones.
- *Recursividad*. Un enunciado está escrito en términos de si mismo.

Diseño

- 1 Paradigma
- 2 Diseño
- 3 Java
- 4 Bibliografía

Temas

2 Diseño

- Metodología Warnier-Orr

Warnier-Orr

- La metodología Warnier-Orr permite diseñar algoritmos. Viso y Peláez V. 2012
- El proceso de diseño estará controlado por los datos, es decir, las estructuras de control están dadas por las estructuras que guardan los datos.
- El diseño parte del estado final del programa (lo que deseamos obtener).
- Va definiendo pequeños pasos que van transformando a los datos hacia el estado inicial del problema (lo que sabemos al inicio).
- Los grupos de datos se organizan en 4 formas, al igual que las estructuras de control:
 - 1 Secuenciación
 - 2 Condicional
 - 3 Repetición
 - 4 *Recursión

Secuenciación

Secuenciación

$$\text{nombre} = \begin{cases} \text{descr}_1 \\ \text{descr}_2 \\ \dots \\ \text{descr}_n \end{cases} \quad (1)$$

Condicional

Condicional

$$\text{nombre} = \left\{ \begin{array}{l} \text{cond}_1 \quad \left\{ \begin{array}{l} \dots \\ \dots \end{array} \right. \\ \oplus \\ \text{cond}_2 \quad \left\{ \begin{array}{l} \dots \\ \dots \end{array} \right. \\ \oplus \\ \dots \\ \oplus \\ \text{cond}_n \quad \left\{ \begin{array}{l} \dots \\ \dots \end{array} \right. \end{array} \right. \quad (2)$$

Repetición

Repetición o *Iteración*

$$\text{nombre} \begin{matrix} \text{(condición)} \end{matrix} = \begin{cases} \text{descr}_1 \\ \text{descr}_2 \\ \dots \\ \text{descr}_n \end{cases} \quad (3)$$

Recursividad

Recursividad

$$\text{nombre} = \begin{cases} \text{descr}_1 & \text{caso base} \\ f(\text{nombre}) & \text{llamada recursiva} \end{cases} \quad (4)$$

Warnier-Orr

$$\text{Nombre del problema} = \left\{ \begin{array}{l} \text{.Principio} \\ \text{Proceso} \\ \text{.Final} \end{array} \right\} \left\{ \begin{array}{l} \text{Obtener datos} \\ \text{Inicializar} \\ \dots \\ \dots \\ \text{Amarrar cabos sueltos} \\ \text{Entregar resultados} \end{array} \right. \quad (5)$$

Java

- 1 Paradigma
- 2 Diseño
- 3 Java**
- 4 Bibliografía

Expresiones y enunciados

- Los comandos en Java se componen a partir de combinar variables, literales y operadores formando *expresiones* que devuelven un valor.
- Los *enunciados* (*statements*) son una composición de expresiones que serán evaluados como una unidad. Los hay de dos tipos:
 - 1 *Enunciados simples* Expresiones que terminan con ;
 - 2 *Enunciados compuestos* aquellos que definen las estructuras de control del lenguaje.

Estructuras de control

- En Java, las estructuras de control se utilizan dentro de *funciones*, para implementar algoritmos.
- Todas las funciones se encuentran dentro de alguna clase.
- Si no usaremos *orientación a objetos*, por el momento declararemos a las funciones como `static`.
- En particular, el punto de entrada `main` es una función estática, no le pertenece a ningún objeto, sólo inicia el algoritmo del programa.

```
1 public class Clase {  
2     public static void main(String[] args) {  
3         System.out.println("; Soy una función!");  
4     }  
5 }
```

Secuenciación

Enunciados <expresión>;

- Composiciones de operaciones primitivas forman *enunciados*.
- En Java todo enunciado termina con punto y coma ;.

```
1 double x = 3.1416 / 2;
```

```
1 f = 2.3f;
```

```
1 System.out.println("Probando");
```

```
1 int n = (x && y) ? 6 / 3 : 4 * 3 * 2;
```

Bloques {}

```
1 <algo al inicio> {  
2     boolean x = true;  
3     boolean y = false;  
4     int n = (x && y) ? 6 / 3 : 4 * 3 * 2;  
5 }
```

- Son una lista de enunciados que serán ejecutados secuencialmente.
- Por *convención* los enunciados dentro de un bloque se escriben *indentados*, es decir, se alinean un número fijo de espacios más a la derecha.
- En Java la llave que abre se escribe en la misma línea del comando que da origen al bloque. Veremos a continuación estas opciones.

Operadores booleanos

Tablas de verdad

OR		
x	y	$x \parallel y$
false	false	false
false	true	true
true	false	true
true	true	true

AND		
x	y	$(x \&\& y)$
false	false	false
false	true	false
true	false	false
true	true	true

Corto circuito

- Cuando el resultado de una operación booleana compuesta se puede deducir resolviendo sólo la primera operación, la segunda ya no se evalúa.
 - `false && true`
Como `false` hace que la operación `&&` sea `false`, el `true` final no se lee.
 - `true || false`
Como `true` hace que el `||` de `true`, el fase final ya no se lee.
- Esto es útil en expresiones más complejas, donde no sabemos si el operando final se puede evaluar o no.
 - `num1 != 0 && num2/num1 > 10`
La operación `num2/num1`, necesaria para verificar la segunda condición, sólo se llevará a cabo si `num1 != 0` es verdadera.

Condicionales

Estructuras de control condicionales

if else

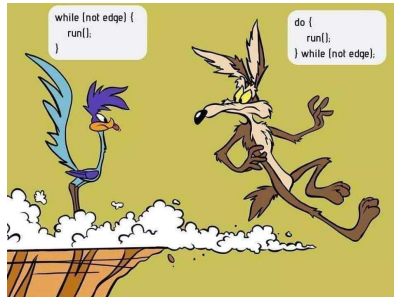
switch case, break, default

Repetición

while break, continue

do while break, continue



for break, continue



Bibliografía

- 1 Paradigma
- 2 Diseño
- 3 Java
- 4 Bibliografía**

Bibliografía I

-  López Román, Leobardo (2011). *Programación estructurada y orientada a objetos. Un enfoque algorítmico*. 3.^a ed. Alfaomega.
-  Viso, Elisa y Canek Peláez V. (2012). *Introducción a las ciencias de la computación con Java*. 2a. Temas de computación. Las prensas de ciencias. 571 págs. ISBN: 978-607-02-3345-6.

Licencia

Creative Commons
Atribución-No Comercial-Compartir Igual

