

Tema 25: Programación Estructurada Estructuras básicas. Funciones y procedimientos

1. Introducción

- Programación Estructurada: paradigma de programación que utiliza estructuras de control de flujo para organizar y manejar la ejecución de un programa.
- Fundamental para tecnologías modernas, dispositivos móviles y ordenadores actuales
- Forma parte del temario oficial de acceso a especialidades de Informática.
- Ubicado en el bloque de "Algoritmos y Programación".
- Importancia en el currículo educativo y tecnológico.

2. Conceptos Previos

- **Informática:** busca soluciones a problemas mediante algoritmos.
- **Programación:** traducción de algoritmos a un lenguaje entendible por la computadora.
- **Lenguaje de programación:** conjunto de reglas y símbolos para escribir programas.
- **Evolución:** de código "espagueti" a paradigmas estructurados y modulares.
- **Paradigmas:** Programación estructurada y modular

3. Programación Estructurada

- Técnicas para desarrollar algoritmos legibles y modificables.
- **Características de un programa propio:**
 - Un solo punto de entrada y salida.
 - Todas las acciones son accesibles.
 - No tiene bucles infinitos.
- **Teorema de Böhm y Jacopini (1966):** los programas pueden construirse solo con estructuras secuenciales, selectivas y repetitivas.

4. Estructuras Básicas

4.1. Estructuras Secuenciales

- Ejecución en orden físico.
- **Ejemplo en Java:**

```
sentencia1;  
sentencia2;
```

4.2. Estructuras Selectivas

- **Condición determina ejecución:**
 - **Simple:** `if (condición) { acciones }`
 - **Doble:** `if (condición) { acciones1 } else { acciones2 }`

- **Múltiple:** `switch (variable) { case valor1: acciones; break; }`

4.3. Estructuras Repetitivas

- **Mientras:** `while (condición) { acciones }`
- **Repetir hasta:** `do { acciones } while (!condición);`
- **Desde o Para:** `for (inicio; condición; iteración) { acciones }`

5. Funciones y Procedimientos

5.1. Funciones

- **Subprogramas que retornan un valor.** 2 tipos, de usuario o de sistema
 - **Declaración:** 2 partes:
 - Cabecera: nombre y parámetros.
 - Cuerpo: instrucciones a realizar por la función.
 - **Invocación:** Llamar la función con los valores necesarios.
- **Ejemplo:**

```
public static int factorial(int n) {  
    int fact = 1;  
    for(int i = 1; i <= n; i++)  
        fact *= i;  
    return fact;  
}
```

5.2. Procedimientos

- **Subprogramas que NO retornan un valor.**
- Declaración e invocación como en las funciones.
- **Ejemplo en pseudocódigo:**

```
procedimiento mostrarMensaje() {  
    imprimir("Hola, mundo");  
}
```

5.3. Paso de Parámetros

- **Por Valor:** el subprograma recibe una copia del dato.
- **Por Referencia:** el subprograma modifica la variable original.

5.4. Ámbito de un Identificador

- **Local:** dentro del subprograma.
- **Global:** accesible desde todo el programa.

6. Recursos y Herramientas

- **Sololearn:** plataforma educativa de programación.
- **Visual Studio Code:** editor de código con soporte para Java y otros lenguajes.
- **Eclipse:** IDE para Java.

7. Aplicaciones en el Contexto Escolar y Laboral

Escolar

- Enseñanza en **Java, Python, C, C++** en ciclos formativos.

Laboral

- **Desarrollo de Software:** Python y Java en aplicaciones web/móviles.
- **Ciencia de Datos e IA:** Python en análisis de datos y machine learning.

8. Conclusión

- **La programación estructurada mejora la legibilidad y mantenimiento del código.**
- **Uso de estructuras de control y subprogramas para optimizar el desarrollo.**

9. Bibliografía

- Joyanes, L. (2020). Fundamentos de programación.
- Prieto, A. (2006). Introducción a la informática.
- López, L. (2004). Programación Estructurada.