

---

# Algoritmos y Estructuras de Datos

## Tema 2: Diseño de Algoritmos

Algoritmos y Estructuras de datos DIT-UPM

1

## Contenidos

---

- 1. Algoritmos recursivos
  - 1.1 Algoritmos recursivos. Recursión simple
  - 1.2 Algoritmos con vuelta atrás y ejemplos
- 2. Complejidad de los algoritmos
- 3. Algoritmos de búsqueda y su complejidad

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Recursividad

- Una **definición** de un **concepto** es **recursiva**, si es una definición que **emplea el propio concepto** en la **definición**
- Por ejemplo: *Máximo Común Divisor*

$$MCD(m,n)= \begin{cases} n, & \text{si } RESTO(m,n) = 0 \\ MCD(n, RESTO(m,n)) & \end{cases}$$

- Un **método** Java es **recursivo** si su cuerpo incluye **llamadas a si mismo**
  - La recursión es **indirecta** cuando un método *m* incluye **llamadas a métodos** que directamente o **indirectamente** incluyen llamadas a *m*

## Ejemplos de métodos recursivos

- Una definición matemática de factorial es:

$$\text{factorial}(n) \text{ es } \begin{cases} 1, & \text{si } n \leq 1 \\ n * \text{factorial}(n-1), & \text{en otros casos} \end{cases}$$

- Una implementación de esa definición en Java es:

```
long factorial(long n) {  
    if (n <= 1) return 1;  
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Estructura de la recursión simple

- La **llamada recursiva** debe estar en el cuerpo de algún tipo de **condición**. Un método recursivo sencillo incluye:

- **Cálculos generales** del método.
- **Ejecuciones base** del método: esas ejecuciones no hay llamadas recursivas. Siempre debe haber al menos una.
- **Ejecuciones recursiva** del método: ejecuciones que producen llamadas recursivas

```
long factorial(long n) {  
    if (n <= 1) return 1;  
    else return n * factorial(n - 1);  
}
```

## Variables en ejecución

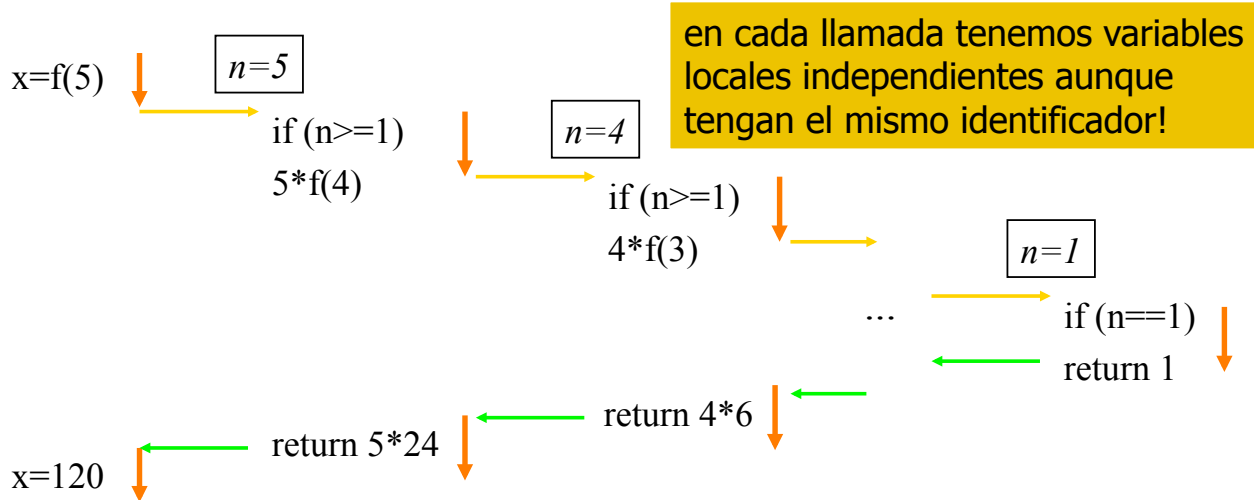
- Variables a las que accede un método recursivo:
  - **Variables locales y parámetros**: cada llamada recursiva tiene su copia de estas variables, y no puede acceder a los valores de las llamadas anteriores
    - Al final de la ejecución se **destruyen** esas variables
    - Mientras está pendiente de resolverse una llamada recursiva, sus llamadas anteriores también están **pendientes** de terminar

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

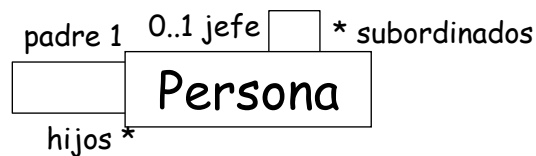
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Ámbito dinámico (recursivo)



## Estructuras recursivas

- Muchas estructuras de datos incluyen **referencias** a objetos de su **misma clase**



- Algunos conjuntos de los objetos crean

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

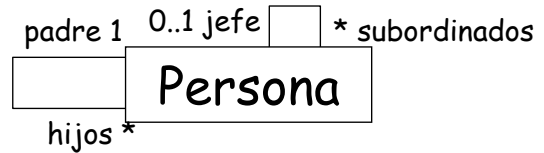
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Estructuras recursivas

```
class Persona {
    String nombre;
    Persona padre;
    Persona[] hijos;
    Persona jefe;
    Persona[] subordinados;
    Persona jefeEmpresa() {

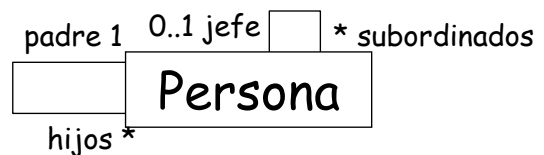
    }
    String descendientes() {

    }
}
```



# Estructuras recursivas

```
class Persona {
    String nombre;
    Persona padre;
    Persona[] hijos;
    Persona jefe;
    Persona[] subordinados;
    Persona jefeEmpresa() {
        if (jefe == null) return this;
        else return jefe.jefeEmpresa();
    }
}
```



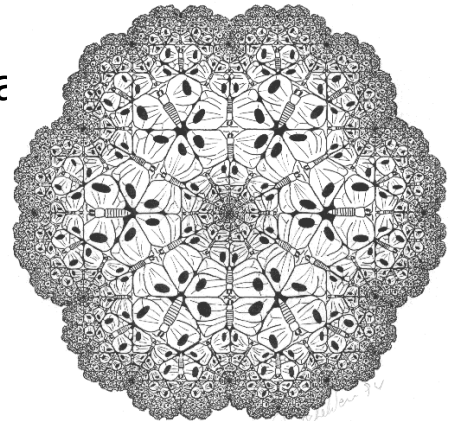
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

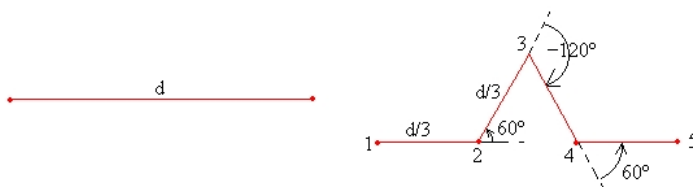
# Curvas recursivas: Fractales

- Un **fractal** es un objeto geométrico cuya **estructura básica**, fragmentada o irregular, se **repite** a diferentes escalas
- El método traza una forma geométrica simple
  - En la **siguiente escala** partes de esa figura pueden representarse con la forma geométrica
    - En la **siguiente escala** partes de esa figura pueden representarse con la forma geométrica
      - ...



## Fractales: curva de Koch

- En el copo de nieve de Koch cada segmento es sustituido por cuatro segmentos, cada uno de ellos de un tercio de la longitud del anterior



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

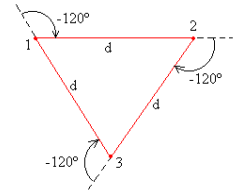
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Fractales: curva de Koch

```
public class KochSnowflake extends GraphicsProgram {
    private static final double FRACCION = 0.75;
    private static final int ORDEN = 5;
    private GPen lapiz;

    public void run() {
        double ancho = 500; // getWidth();
        double alto = 500; // getHeight();
        lapiz = new GPen();
        add(lapiz, ancho / 2, alto / 2);
        dibujaFractalKoch(FRACCION * Math.min(ancho, alto), ORDEN);
    }
    private void dibujaFractalKoch(double extremo, int orden) {
        lapiz.move(-extremo / 2, -extremo / (2 * Math.sqrt(3)));
        dibujaLineaFractal(extremo, 0, orden);
        dibujaLineaFractal(extremo, -120, orden);
        dibujaLineaFractal(extremo, +120, orden);
    }
}
```



Algoritmos y Estructuras de datos DIT-UPM

13

# Fractales: curva de Koch

```
private void dibujaLineaFractal(double r, int theta, int orden) {
    if (orden == 0) {
        lapiz.drawPolarLine(r, theta);
    } else {
        dibujaLineaFractal(r / 3, theta, orden - 1);
        dibujaLineaFractal(r / 3, theta + 60, orden - 1);
        dibujaLineaFractal(r / 3, theta - 60, orden - 1);
        dibujaLineaFractal(r / 3, theta, orden - 1);
    }
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Llamadas recursivas

- Cada llamada tiene asociado un **retorno**
  - Una **excepción** aborta solo la **llamada**
    - Las excepciones pueden seguir el anidamiento
- La **primera llamada en terminar** será la **última** llamada hecha, y la **última** en terminar será la **primera**
- La recursividad debe terminar en algún momento y el **número de llamadas anidadas** debe ser **limitado**
  - Si no hay memoria suficiente -> excepción
- La(s) llamada(s) recursiva(s) debe(n) estar en una sentencia alternativa o en el cuerpo de un bucle que no siempre se ejecuta

# Recursividad vs bucle

- **Recursividad** es una forma de **iteración**
- Dos formas alternativas de resolver un problema

```
int fact(int n) throws Exception {
    if (n < 0) throw new Exception("argumento no válido");
    int f=1;
    while (n > 1) {
        f *= n;
        n--;
    }
    return f;
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70



# Recursividad vs bucle

- Un **bucle** podemos implementarlo con sentencias condicionales (if) y **recursividad**:

```
void rellenar(int[] a, int desde) {  
    for (int i=desde; i >= 0; i--)  
        a[i]=i;  
}
```

```
void rellenar(int[ ] a, int n) {  
    if (n < 0) return;  
    else {  
        a[n] = n;  
        rellenar(a, n - 1);  
    }  
}
```

Ejecución Base

Cálculos Generales

Ejecución Recursiva

Algoritmos y Estructuras de datos DIT-UPM

17

## Método recursivo y fachada

- Las implementaciones de los métodos recursivos suelen emplear **parámetros auxiliares** para controlar los niveles de recursividad
  - Esos parámetros crean confusión (no sabemos que poner) para quién usa el método, porque desconocemos la implementación
- Un **método fachada** es la entrada pública, un método privado recursivo implementa la operación

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

Algoritmos y Estructuras de datos DIT-UPM

18

Cartagena99