Métodos y funciones

PROGRAMACION I

Grado en Matemáticas e Informática

TAXONOMÍA DE PROBLEMAS

Solución Directa: El algoritmo se especifica a través de una fórmula. Se representa con una expresión

Análisis de casos: El algoritmo tiene que distinguir entre varios casos posibles. Se representa con construcciones sintácticas de selección de alternativas

Recorrido: El algoritmo requiere realizar una recombinación de cálculos. Se representa con una construcción sintáctica de repetición

PROBLEMAS COMO FUNCIONES

- Las funciones ofrecen una sintaxis clara y conocida
- Tienen un nombre
- Hacen mención a los datos que manejan
- Permiten expresar cálculos
- Ofrecen un resultado

FUNCIONES EN Java

- Formato: Cabecera y cuerpo

```
public static <<TipoRes>> <<Nombre>> (<<ListaParámetros>>)
{
  <<Bloque>>
}
```

"{" y "}" comienzo y fin del cuerpo de la función

FUNCIONES EN Java

TipoRes: Dominio (tipo) del resultado

Nombre: Identificador que da nombre a la función. Debe empezar por una letra minúscula.

ListaParámetros: Secuencia de pares `TipoParametro NombreParametro` separados por “`,`”

Bloque: Secuencia de órdenes (sentencias) separadas por “`;`”

FUNCIONES EN Java

- Sentencia return
- Formato: `return <<expresión>>;`
- Funcionamiento:
  - Se evalúa la expresión. El valor será el resultado de la función.
  - Se termina la función.
- Obligatoria en el bloque de una función.

VISIBILIDAD

- Todo lo declarado en el cuerpo de la función tiene consideración local
- Los parámetros formales quedan declarados en la cabecera. También tienen consideración local
- Lo local es invisible desde fuera

EJEMPLOS DE CODIFICACIÓN

Área de un circulo

```
public static double areaCirculo (double radio)
{
  return Math.PI * radio * radio;
}
```

EJEMPLOS DE CODIFICACIÓN

Volumen de un cilindro

```
public static double volumenCilindro (double radio, double altura)
{
  return altura * areaCirculo (radio);
}
```

EJERCICIOS DE CODIFICACIÓN

Ejercicio4: “Área del triángulo”

Ejercicio5: “Ser un número múltiplo de otro”

Ejercicio6: “Menor de dos números”

PARÁMETROS

- Declaración de una función: El código completo de la función (cabecera y cuerpo)
- Invocación de una función: El nombre seguido de la lista de parámetros actuales
- Parámetros formales: Los que aparecen en la declaración
- Parámetros actuales: Los que aparecen en la invocación

PARÁMETROS

```
public static int cubo (int numero) {
  return numero * numero * numero;
}
```

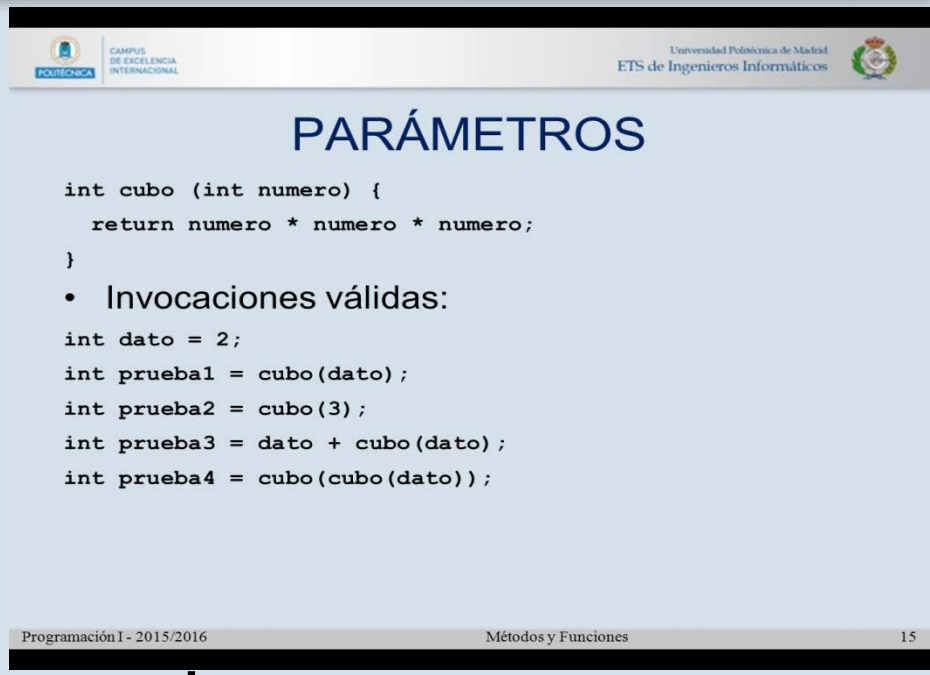
- Invocaciones válidas:

```
int dato = 2;
int prueba1 = cubo(dato);
int prueba2 = cubo(3);
int prueba3 = dato + cubo(dato);
int prueba4 = cubo(cubo(dato));
```

PARÁMETROS

Parámetro formal de `cubo`:

`numero`



Parámetros actuales en cada prueba:

En la 1: `dato`

En la 2: `3`

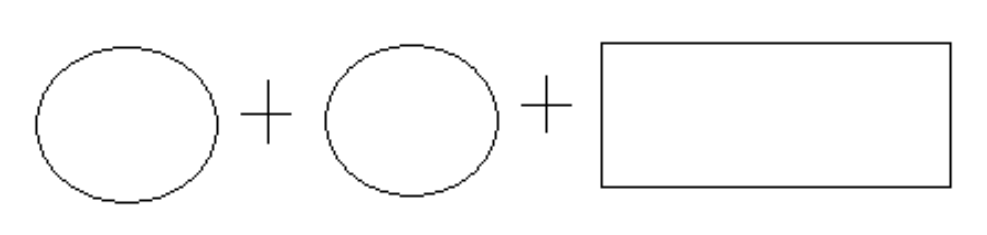
En la 3: `dato`

En la 4: `cubo(dato)`

EJERCICIO

Problema: “Superficie o área total del cilindro”

- La superficie comprende las dos bases, que son círculos, más el área lateral o área del contorno, que es un rectángulo.



FUNCIONES NECESARIAS

```
public static double areaRectangulo (double base,
                                     double altura)
{
  return base * altura;
}
```

FUNCIONES NECESARIAS

```
public static double longitudCircunferencia (double radio)
{
  return 2 * Math.PI * radio;
}
```