

CHULETA SEXTA VUELTA ESPIRAL

| ELEMENTO DEFINIDO | SINTAXIS EN JAVA | | |
|------------------------------|---|---|---|
| Paquete | Package nombre.del.paquete; | | |
| Clase | import paquete.Clase; import paquete.*; public class NombreClase { // Propiedades // Métodos } | | |
| Propiedades | <i>Una sola propiedad:</i> public tipoDato nombrePropiedad1; <i>Varias propiedades del mismo tipo en una sola línea:</i> public tipoDato propiedad1, propiedad2; | | |
| Los tipos de datos | Ver en la otra carilla del folio | | |
| Métodos | public tipoDato nombreMetodo (tipoDato par1, ...) { // instrucciones de un método return valorCalculado; } public void nombreMetodo (tipoDato par1, ...) { // instrucciones del método (no hay return) } | | |
| Instrucciones de un método | <i>Definición y creación de un objeto en una línea:</i> NombreClase nombreObj = new NombreClase(); | <i>Uso de los métodos un objeto:</i> nombreObj.nombreMetodo (valor1, valor2, ...); | <i>Declaración de variables:</i> tipoDato nombreVariable; |
| | <i>Asignación:</i> Propiedad = expresión; Variable = expresión; | <i>Lectura del teclado:</i> Scanner sc = new Scanner(System.in); int valorLeido = sc.nextInt(); | <i>Imprimir en pantalla:</i> System.out.println("Texto"+elemValido+"Texto"+...); System.out.printf("Texto y %algo", param...); |
| | <i>Condicional simple:</i> if (exprBool) { //instrucciones } | <i>Condicional doble:</i> if (exprBool) { //instrucciones } else { //instrucciones } | <i>Condicional múltiple:</i> switch (exprSw) { caso literal1: //instrucciones break; caso literal2: ... default: ... } |
| | <i>Bucle do... while:</i> do { // Cuerpo bucle } while (ExprBool); | <i>Bucle while:</i> while (ExprBool) { // Cuerpo bucle } | <i>Bucle for:</i> for (i = 0; ExprBool; i++) { //Cuerpo bucle } |
| Expresión | elemValido operador elemValido operador... | | |
| Elemento válido del lenguaje | <i>Literales:</i> 4, 5, 230, 'A', 34.34... <i>Variables o Propiedades</i> <i>Constantes:</i> final tipoDato NOMBRE = valor; | | |

| | |
|---|---|
| Operadores | Aritméticos: $+, -, *, /, (,), \%, ++$ y $-$ Relacionales: $<, >, >=, <=, ==, !=$ Lógicos: $\&\&, , !$ |
| ExprBool y ExprSw | <i>exprBool</i> es una expresión cuyo resultado es un boolean <i>exprSw</i> es una expresión cuyo resultado es un int o un String |
| ClasePrincipal (contiene el método main) | <pre>public class PruebaNombreClase { public static void main (String args[]) { // instrucciones de un método } }</pre> |

PRIORIDAD DE OPERADORES

| | Descripción | Operadores |
|---------------|----------------------------------|-----------------|
| + PRIORITARIO | Paréntesis | () |
| | operadores incremento/decremento | op++ op-- |
| | operadores unarios | +op -op ! |
| | Multiplicación, división y resto | * / % |
| | suma y resta | + - |
| | operadores relacionales | < > <= >= == != |
| | AND booleano | && |
| | OR booleano | |
| - PRIORITARIO | operador de asignación | = |

| TIPOS DE DATOS | CLASIFICACIÓN | NOMBRE | TIPO | OCUPA EN MEMORIA | RANGO APROXIMADO | LITERALES |
|----------------|--|---|--|------------------|------------------------------------|---------------------|
| | TIPOS PRIMITIVOS (elementos simples como números o caracteres) | byte | Número entero | 1 byte | -128 a 127 | (byte) 0... |
| | | short | Número entero | 2 bytes | -32.768 a 32.767 | (short) 0... |
| | | int | Número entero | 4 bytes | -2×10^9 a 2×10^9 | 2, 323, -12... |
| | | long | Número entero | 8 bytes | Muy grande | 2L, 323L, -12L... |
| | | float | Número real (precisión normal) | 4 bytes | Muy grande | 2.1f, -12.324f... |
| | | double | Número real (precisión doble) | 8 bytes | Muy grande | 2.1, -12.324... |
| | | char | Carácter simple | 2 bytes | --- | A, 'a', '?', 'R'... |
| | | boolean | Valor true o false | 1 byte | --- | true, false |
| | TIPOS OBJETO (con métodos, necesitan una invocación para ser creados) | Tipos de la biblioteca estándar de Java | String (cadenas de texto)... | | | "Hola", "Manuel" |
| | | Tipos definidos por el programador/a | Cualquier clase que programemos: Mechero, Bombilla, Coche, TarjetaCredito... | | | N/A |

NOTA: En **negrita** los más habituales