# UD1. IDENTIFICACIÓN de los elementos de un programa INFORMÁTICO

# BOLETÍN

1. Marca con una X la herramienta mínima necesaria para poder realizar las operaciones que se indican en la tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ***Herramientas*** | | | |
| ***Operación*** | **JVM** | **JRE** | **JDB** | **JDK** |
| Ejecutar una aplicación Java | **x** | **x** |  |  |
| Desarrollar aplicaciones Java | **x** | **x** | **x** | **x** |

1. Marca con una X la extensión correspondiente a cada tipo de fichero:

|  |  |  |
| --- | --- | --- |
|  | ***Extensión*** | |
| ***Tipo de fichero*** | **java** | **class** |
| Fichero compilado de Java |  | **x** |
| Fichero fuente de Java | **x** |  |

1. Indica la afirmación correcta:
2. Los ficheros compilados en Java están codificados en el lenguaje bytecode que puede ser interpretado por cualquier sistema operativo de forma directa. **Falsa**
3. Los ficheros compilados en Java están codificados en el lenguaje máquina para que puedan ser ejecutados por un sistema operativo concreto. **Falsa**
4. Los ficheros compilados en Java están codificados en el lenguaje máquina común a Linux y Windows para que puedan ser ejecutados cualquiera de estos sistemas operativos. **Falsa**
5. Los ficheros compilados en Java están codificados en el lenguaje bytecode para que puedan ser ejecutados por la JVM que debe estar instalada en el sistema. **Correcta**
6. Si suponemos que tenemos el JDK almacenado en la siguiente ruta C:\Program Files\Java\jdk-11.0.8, indica que ruta debemos de asignar a la variable JAVA\_HOME.

**C:\Program Files\Java\jdk-11.0.8 (ya que es esta ruta está la carpeta bin)**

1. Si tenemos un fichero mifichero.java y ejecutamos, desde la línea de comandos:

**javac mifichero.java**

* 1. ¿Cómo se llama la operación que realiza? **Compilación**
  2. ¿Qué ficheros se generan en la operación? **Ficheros objeto tipo class**
  3. De los ficheros que existen después de ejecutar javac mifichero.java indica cuál es el fichero fuente y el fichero objeto y en qué lenguaje está codificado el contenido de cada uno.

**Fichero fuente: mifichero.java / Su contenido está codificado en lenguaje java.**

**Fichero objeto: mifichero.class /Su contenido está codificado en bytecode.**

* 1. Indica cuál sería la orden que tendríamos que escribir en la línea de comandos para ejecutar el fichero resultante de la operación.

**java mifichero (desde el path en el que se encuentra el fichero)**

1. Indica cuales de las siguientes afirmaciones son verdaderas.

|  |  |
| --- | --- |
| Operación | ¿Verdadera? |
| Todo proyecto Java tiene que tener como mínimo una clase que debe contener un método *main* | Verdadera |
| Los IDE para Java utilizan internamente las herramientas básicas de JVM para realizar las operaciones descritas | Falso |
| Un entorno de desarrollo integrado o IDE (Integrated Development Environment) es una aplicación que facilita el desarrollo de aplicaciones en algún lenguaje de programación | Verdadero |
| El método main es private | Falso |
| El método main es estático | Verdadero |
| El método main, como resultado de su ejecución devuelve o valor boolean | Falso |

1. Indica las características de los comentarios que se indican en la tabla:

|  |  |
| --- | --- |
| Símbolo de comienzo | Características de los comentarios |
| //comentario | Comentario de una línea |
| /\*comentario\*/ | Comentarios de bloques. De más de una línea |
| /\*\*comentario\*/ | Documentación de referencia del paquete (que utiliza javadoc) |

1. Indica si el identificador es válido en cada caso, si no lo es, indica por qué.

|  |  |  |
| --- | --- | --- |
| Identificador | Válido/No válido | Explicación en el caso de que no sea válido |
| MiVAriable | Válido |  |
| MIVARIABLE | Válido |  |
| Mi Variable | No valido | No se pueden emplear espacios |
| 123Java | No válido | No puede empezar por un dígito |
| X | Válido |  |
| variable-2 | No válido | No se pude usar el carácter - |
| X1 | Válido |  |
| a+c | No válido | No se puede usar el carácter + |
| class | No válido | Se trata de una palabra reservada |
| \_mivariable | Válido |  |
| $mivariable | Válido |  |
| suma\_de\_array | Válido |  |
| aprenderJava | Válido |  |
| suma&resta | No valido | No se puede usar el carácter & |
| true |  |  |
| suma.de.array | No válido | No se puede usar el carácter . |

1. Indica de qué tipos son los literales siguientes y señala aquellos que no son correctos.

|  |  |  |
| --- | --- | --- |
| Literal | Tipo | Comentarios |
| 982 | int |  |
| 1234L | long |  |
| 0b0100101 | int |  |
| ‘a’ | char |  |
| 12.565 | double |  |
| “Tolosa” | String |  |
| 0.354 | double |  |
| 154\_.657 | double | No es correcto. No se puede usar \_ alrededor de un punto decimal. |
| +1200 | **int** |  |
| ‘5’ | char |  |
| .006E4 | double |  |
| “ ” | String |  |
| 0b1010000101000101101000 | **int** |  |
| ‘\n’ | char |  |
| 34,56L |  | No es ni long ni double. Si fuera long (indicado por la L al final) no podría ser decimal y si fuera double, el decimal se marca con un punto. |
| 234\_456\_345L | long |  |
| -25 | int |  |
| ‘\u0061’ | char |  |
| 02344 | int |  |
| 1234 | int |  |
| .87 | double |  |
| 2.34E3 | double |  |
| +74.356E-3F | float |  |

1. Si tenemos la expresión a>b indica si es cierto o falso las siguientes afirmaciones. Probar en el ordenador.

|  |  |
| --- | --- |
| Afirmación | Cierto |
| a y b pueden ser de tipo char | Cierto |
| a y b pueden ser de tipo String | Falso |
| a y b pueden ser de tipo float | Cierto |
| a y b pueden ser de tipo boolean | Falso |

1. Para hacer en el papel: Indica que muestra por pantalla este código:

int a = 20, b = 10, c = 0, d = 20, e = 40, f = 30;

boolean condition = true;

c = ++a;

System.out.println("Valor de c (++a) = " + c);

c = b++;

System.out.println("Valor de c (b++) = " + c);

c = --d;

System.out.println("Valor de c (--d) = " + c);

c = e--;

System.out.println("Valor de c (e--) = " + c);

System.out.println("Valor de !condition = " + !condition);

**Valor resultante de c (++a) = 21**

**Valor resultante de c (b++) = 10**

**Valor de c (--d) = 19**

**Valor de c (e--) = 40**

**Valor de !condition = false**

1. Implementar un proyecto que, dado un número de horas, calcule el número de semanas, días y horas equivalentes. Por ejemplo, para 1000 horas debe mostrar 5 semanas, 6 días y 16 horas.

**public static void main(String[] args) {**

**int numero;**

**Scanner input = new Scanner(System.in);**

**System.out.println("Introduce un número (entero) de horas: ");**

**numero = input.nextInt();**

**int horas=(numero%24);**

**int numero\_dias=(numero/24);**

**int dias=(numero\_dias%7);**

**int semanas=(numero\_dias/7);**

**System.out.println(numero+" horas son "+semanas+" semana(s),"+dias+" día(s) y "+horas+" hora(s).");**

**}**