Java Teórico

1.1. Indicar en cuál de las siguientes empresas tiene su origen el lenguaje de programación Java:

1. Sun Microsystems
2. IBM
3. Microsoft
4. Dell Computers
5. Hewlett-Packard

1.2. Indicar el nombre del creador del lenguaje Java:

1. Bill Gates
2. Linus Torvalds
3. James Gosling
4. Tim Berners-Lee
5. Ninguno de los anteriores

1.3. Indicar cuál de las siguientes extensiones corresponde al archivo fuente de un programa escrito en Java:

1. .font
2. .java
3. .class
4. .exe

1.5. Para la ejecución de un programa de Java previamente compilado se emplea...

1. un intérprete
2. un editor de texto
3. un compilador
4. un depurador
5. un compresor

2.1. Indicar cuál o cuáles de las siguientes lineas de código corresponden a un comentario en Java:

1. { Esto es un comentario }
2. (\* Esto es un comentario )
3. /\* Esto es un comentario /
4. / Esto es un comentario \*/
5. // Esto es un comentario
6. Ninguna de las anteriores

2.2. Indicar cuáles de los siguientes caracteres puede utilizarse para componer un identificador en el código fuente de un programa en Java

1. el carácter de subrayado: \_
2. el carácter de espacio en blanco
3. el carácter de la letra mayúscula Z
4. el carácter de guión: -
5. Ninguno de los anteriores

2.3. La secuencia de caracteres que componen un identificador en el código fuente de Java...

1. debe empezar por una letra minúscula o un carácter de subrayado
2. puede empezar por cualquier carácter ASCII
3. debe empezar por una letra o un dígito decimal
4. debe empezar por una letra mayúscula
5. Ninguna de las anteriores

2.4. Indicar cuál o cuáles de los siguientes identificadores son válidos en el codigo fuente de un programa en Java:

1. PrimerValor
2. Valor2
3. 4Valor
4. valor
5. Tercer\_Valor
6. Ninguno de los anteriores

2.5. El tamaño o longitud de la secuencia de caracteres correspondiente a un identificador en el código fuente de un programa en Java...

1. debe estar comprendido entre 2 y 64 caracteres
2. debe ser de 16 caracteres como máximo
3. debe ser de 32 caracteres como máximo
4. debe ser de 2 caracteres como mínimo
5. Ninguna de las anteriores

3.1. Indicar cuál es el intervalo de representación del tipo primitivo short de Java:

1. -128..127
2. 0..65536
3. -32768..32767
4. 0..256
5. ninguno de los anteriores

3.2. Indicar cuál o cuáles de los siguientes valores o constantes literales puede asignarse a una variable de tipo boolean:

1. boolean
2. falso
3. false
4. "false"
5. not
6. Ninguno de los anteriores

3.3. Indicar el espacio utilizado por un dato de tipo primitivo int en Java:

1. 2 bytes
2. 6 bytes
3. 4 bytes
4. 1 byte
5. ninguno de los anteriores

3.4. Indicar cuál o cuáles de los siguientes tipos primitivos de Java utiliza el formato en punto o coma flotante para representar cualquiera de sus posibles valores:

1. El tipo primitivo short
2. El tipo primitivo int
3. El tipo primitivo double
4. El tipo primitivo long
5. Ninguno de los anteriores

3.5. Indicar cuántas veces se le puede asignar un valor a una variable de un programa:

1. Un número indefinido de veces, dentro del intervalo de representación correspondiente
2. Una sola vez, en el cuerpo del programa
3. Solamente una vez, en la sentencia de declaración de la variable
4. Dos veces. Una en la declaración y otra en el cuerpo del programa
5. Ninguna de las anteriores

3.6. ¿Cuál es la palabra reservada que se emplea en la sentencia de declaración de una constante en el código fuente de un programa en Java?

R: final

3.7. ¿Cuál es el identificador del tipo primitivo de Java que permite representar un dato de tipo carácter según el sistema de codificación UNICODE?

R: la comilla simple (')

3.8. Indicar, en número de bytes, el tamaño del espacio reservado en memoria para una variable de tipo double.

R: El espacio reservado en memoria para un variable tipo double es de 8 bytes

4.1. Se quiere almacenar en una variable c de tipo int el resultado de la división de dos variables a y b de tipo int. Indicar cuál o cuáles operadores de Java podrían utilizarse en la expresión del término de la derecha de la asignación c = a operador b:

1. div
2. /
3. %
4. mod
5. Ninguno de los anteriores

4.2. Indicar cuál o cuáles de los siguientes operadores tiene mayor nivel de prioridad en Java:

1. \*
2. +
3. ==
4. ||
5. =

4.3. Indicar cuál o cuáles de los siguientes operadores son binarios (tienen dos operandos):

1. ++
2. ==
3. &&
4. \*
5. Ninguno de los anteriores

4.4. Dada una constante PI con el valor real 3.1415 y una variable r de tipo double, indicar cuál o cuáles de las siguientes expresiones representan la superficie de un semicírculo de radio r:

1. (PI/2)rr
2. PI/2rr
3. (PIrr)/2
4. PIrr/2
5. Ninguna de las anteriores

4.5. Indicar cuál o cuáles de los siguientes son operadores de relación de Java:

1. :=
2. !=
3. \*=
4. ==
5. Ninguno de los anteriores

4.6. Considerando que i es el identificador de una variable entera, indicar el tipo de dato correspondiente a la evaluación de la expresión: i==0

1. el resultado es de tipo carácter
2. es una sentencia de asignación (asigna el valor 0 a la variable entera i)
3. el resultado es de tipo lógico o booleano
4. el resultado es de tipo entero
5. Ninguna de las anteriores

5.1. En el código fuente de un programa escrito en Java, en general las sentencias se separan unas de otras...

1. separándolas con comentarios
2. utilizando un carácter de punto y coma (;)
3. escribiéndolas en distintas líneas
4. separándolas con guiones (-)
5. Ninguna de las anteriores

5.2. En Java, inmediatamente detrás de if (expresion\_booleana) aparece siempre:

1. then sentencia
2. do sentencia
3. begin sentencia
4. else sentencia
5. Ninguna de las anteriores

5.3. Indicar cuales de las siguientes palabras reservadas pueden encontrarse en una sentencia condicional multirramificada o de selección múltiple de Java:

1. default
2. break
3. switch
4. case
5. Ninguna de las anteriores

5.4. Indicar a qué tipo de sentencias corresponde el flujograma de la figura:

1. Sentencia for ...
2. Sentencia if ... else ...
3. Sentencia while ...
4. Sentencia do ... while ...
5. Ninguna de las anteriores

6.1. ¿Qué palabra reservada acompaña siempre a do en el bucle correspondiente?

R: la palabra reservada que acompaña siempre al do en el bucle correspondiente es "while".

6.2. Indicar la salida por pantalla del siguiente programa:

public class b\_while; {

public static void main (String [] args) {

int i=1;

int j=8;

while (i<5) {

i++;

j++;

}

System.out.println(j);

}

}

R: 8

9

10

11

6.3. Indicar con cuáles de los siguientes grupos de palabras reservadas puede construirse un bucle o sentencia repetitiva en Java

1. switch ... case ...
2. for ...
3. while ...
4. do ... while ...
5. if ... else ...
6. Ninguno de los anteriores

6.4. Indicar a qué tipo de sentencias corresponde el flujograma de la figura:

1. Sentencia while ...
2. Sentencia switch...
3. Sentencia if ... else ...
4. Sentencia do ... while ...
5. Ninguna de las anteriores

7.1 Para finalizar (o abandonar) la ejecución de una sentencia repetitiva o bucle puede incluirse dentro de la sentencia que se repite...

1. continue;
2. finally;
3. break;
4. return;
5. Ninguna de las anteriores

7.2 Para detener temporalmente la ejecución de una sentencia repetitiva o bucle puede incluirse dentro de la sentencia que se repite...

1. continue;
2. finally;
3. break;
4. return;
5. Ninguna de las anteriores

7.3. Indicar la salida por pantalla al ejecutar el siguiente programa:

public class TablaProductoA {

public static void main (String [] args) {

int valor;

valor = Integer.parseInt(args[0]);

System.out.println("Tabla de multiplicar del numero " + valor);

for (int i=0; i<11; i++) {

System.out.println(valor + " \* " + i + " = “ + valor\*i );

if (i==4) break;

}

}

}

R: Tabla de multiplicar del numero 3

3 \* 1 = 3

3 \* 2 = 6

3 \* 3 = 9

3 \* 4 = 12

7.4. Indicar la salida por pantalla al ejecutar el siguiente programa:

public class TablaProductoB {

public static void main (String [] args) {

int valor;

valor = Integer.parseInt(args[0]);

System.out.println("Tabla de multiplicar del numero " + valor);

for (int i=0; i<11; i++) {

System.out.println(valor + " \* " + i + " = “ + valor\*i );

if (i==4) continue;

}

}

}

R: Tabla de multiplicar del numero 5

5 \* 1 = 5

5 \* 2 = 10

5 \* 3 = 15

5 \* 5 = 25

5 \* 6 = 30

5 \* 7 = 35

5 \* 8 = 40

5 \* 9 = 45

5 \* 10 = 50

8.1. Indicar cuál o cuáles de los siguientes identificadores de tipos de dato puede corresponder el valor devuelto por un método en Java:

1. int
2. double
3. char
4. boolean
5. Ninguno de los anteriores

8.2. Indicar qué palabra reservada de Java se utiliza en el cuerpo de la declaración de un método para indicar la expresión correspondiente al valor de retorno

R: return

8.3. Indicar qué palabra reservada de Java se utiliza siempre en la cabecera de la declaración de un método que no devuelve ningun valor de retorno.

R: static

8.4. La capacidad mediante la cual dentro del cuerpo de un método puede realizarse una llamada al mismo método se denomina:

1. sobrecarga
2. dinamismo
3. exportabilidad
4. recursión, recurrencia o recursividad
5. Ninguno de los anteriores

8.5. Indicar la salida por pantalla al ejecutar

public class PruebaPar {

public static void main (String [] args){

for (int i=5; i>=-5; i--)

System.out.println(i + " es par: " + esPar(i));

}

public static boolean esPar(int x) {

if (x%2==0) return true;

else return false;

}

}

R: 4 es par: true

3 es par: false

2 es par: true

1 es par: false

0 es par: true

-1 es par: false

-2 espar: true

-3 es par: false

-4 es par: true

-5 es par: false