Java Teórico

Q1.

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

1.2. Indicar el nombre del creador del lenguaje Java: **James Gosling.**

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

1.5. Para la ejecución de un programa de java previamente compilado de emplea: **un interprete**

Q2.

2.1. Indicar cual o cuales de las siguientes líneas de código corresponden a un comentario en Java: **//Esto es un comentario**

2.2. Indicar cuales de los siguientes caracteres puede utilizarse para componer un identificador en el código fuente de un programa en Java: **El carácter de subrayado: \_, el carácter de la letra mayúscula z.**

2.3. La secuencia de caracteres que componen un identificador en el código fuente de Java: **Debe empezar por una letra minúscula o un carácter subrayado.**

2.4. Indicar cual o cuales de los siguientes identificadores son válidos en el código fuente de un programa en Java: **PrimerValor, Valor2, valor, Tercer\_Valor**

2.5. El tamaño o longitud de la secuencia de caracteres correspondiente a un identificador en el código fuente de un programa en Java: **Ninguna de las anteriores, se recomienda evitar las líneas de mas de 80 caracteres.**

Q3.

3.1. Indicar cual es el intervalo de representación del tipo primitivo short de Java: **-32768…32767.**

3.2. Indicar cual o cuales de los siguientes valores o constantes literales puede asignarse a una variable de tipo boolean: **false.**

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

3.4. Indicar cual o cuales de los siguientes tipos primitivos de Java utiliza el formato en punto o coma flotante para representar cualquier de sus posibles valores: **El tipo primitivo double.**

3.5. Indicar cuantas veces se le puede asignar un valor a una variable de un programa: **Un numero indefinido de veces, dentro del intervalo de representación correspondiente.**

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? **La palabra reservada es 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? **El tipo primitivo char.**

3.8. Indicar, en numero de bytes, el tamaño del espacio reservado en memoria para una variable de tipo double: **Se reservan 8 bytes.**

Q4.

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 cual o cuales operadores de Java podrían utilizarse en la expresión del termino de la derecha de la asignación c = a operador b: **c = a/b.**

4.2. Indicar cual o cuales de los siguientes operadores tiene mayor nivel de prioridad en Java: **+**

4.3. Indicar cual o cueles de los siguientes operadores son binarios (tienen dos operandos): **Ninguna de las 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: **PI/2*r*r, (PI*r*r)/2 y PI*r*r/2 representan la superficie.**

4.5. Indicar cuál o cuáles de los siguientes son operadores de relación de Java**: ¡= y ==.**

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: Es de tipo lógico o booleano.**

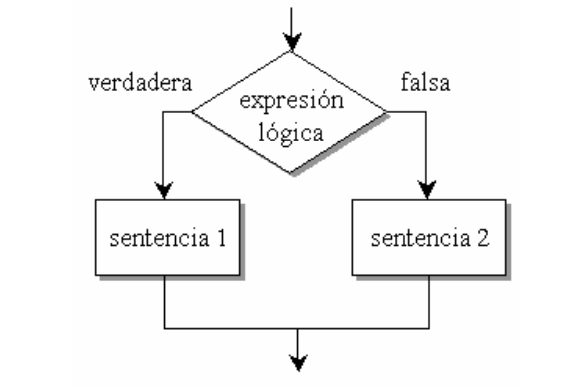
Q5.

5.1. En el código fuente de un programa escrito en Java, en general las sentencias se separan unas de otras... **Utilizando un carácter de punto y como (;).**

5.2. En Java, inmediatamente detrás de if (expresion\_booleana) aparece siempre: **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: **default, break, switch case.**

5.4. Indicar a qué tipo de sentencias corresponde el flujograma de la figura: **Sentencia if…else…**



Q6.

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

**La palabra reservada que siempre acompaña al bucle do es “while”.**

6.2. Indicar la salida por pantalla del siguiente programa: **12**

public class b\_while; {

public static void main (String [] args) {

int i=1;

int j=8;

while (i<5) {

i++;

j++;

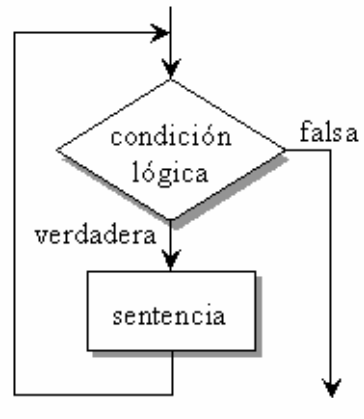
}

System.out.println(j);

}

}

6.3. Indicar con cuáles de los siguientes grupos de palabras reservadas puede construirse un bucle o sentencia repetitiva en Java: **for… while… y do…while… son sentencias repetitivas.**

6.4. Indicar a qué tipo de sentencias corresponde el flujograma de la figura: **Sentencia while…**

Q7.

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

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

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 número " +7);

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

System.out.println(7+ " \* " + i + " = “ + 7\*i );

if (i==4) break;

}

}

}

**La salida por pantalla es:**

**Tabla de multiplicar del número <valor>**

**<valor> \* 0 = 0**

**<valor> \* 1 = <valor>**

**<valor> \* 2 = <valor>\*2**

**<valor> \* 3 = <valor>\*3**

**<valor> \* 4 = <valor>\*4**

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 número " + valor);

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

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

if (i==4) continue;

}

}

}

**La salida por pantalla es:**

**Tabla de multiplicar del número <valor>**

**<valor> \* 0 = 0**

**<valor> \* 1 = <valor>**

**<valor> \* 2 = <valor>\*2**

**<valor> \* 3 = <valor>\*3**

**<valor> \* 4 = <valor>\*5**

**<valor> \* 6 = <valor>\*6**

**<valor> \* 7 = <valor>\*7**

**<valor> \* 8 = <valor>\*8**

**<valor> \* 9 = <valor>\*9**

**<valor> \* 10 = <valor>\*10**

Q8.

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

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: **return**.

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

8.4. La capacidad mediante la cual dentro del cuerpo de un método puede realizarse una llamada al mismo método se denomina: **recursión, recurrencia o recursividad.**

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;

}

}

**La salida por pantalla es:**

**5 es par: false**

**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 es par: true**

**-3 es par: false**

**-4 es par: true**

**-5 es par: false**

***Matias Nicolas Diaz 7º3º***