

Apellidos: _____ Nombre: _____ DNI: _____

Centro Asociado en el que está MATRICULADO: _____ Especialidad: _____

INSTRUCCIONES: Complete sus datos personales en la cabecera de esta hoja, y **ENTRÉGUELA OBLIGATORIAMENTE** con el resto de hojas de su examen. **Cualquier examen que no venga acompañado de esta hoja de enunciados no será corregido.** Complete **TODOS** los datos que se piden en la hoja de lectura óptica o **en caso contrario su examen no será corregido.** El examen consta de 20 preguntas. Cada respuesta acertada tiene un valor de 0.5 puntos y cada respuesta incorrecta de -0.25 puntos. Para superar el examen es necesario obtener una puntuación de al menos 5 puntos.

Test : Conteste exclusivamente en una HOJA DE LECTURA ÓPTICA, no olvidando marcar que su tipo de examen es A.

1.- ¿Cuál sería la salida del siguiente fragmento de código?

```
{int x=12;
    {int x=100;
    }
System.out.println(x);
}
```

- A) 12
- B) 100
- C) null
- D) **Ninguna de las anteriores**

Comentario

No se puede definir dentro de un ámbito otra vez la misma variable

2.- ¿Cuál sería la salida del siguiente fragmento de código?

```
class prueba {
    static int i=50;
}
prueba uno=new prueba();
prueba dos=new prueba();
uno=dos;
uno.i=20;
dos.i=30;
System.out.println(uno.i);
```

- A) 50
- B) 20
- C) **30**
- D) Ninguna de ellas.

Comentario

La salida es este valor ya que al modificar **dos** también se modifica uno debido a que apuntan al mismo sitio

3.- Señale cual de las siguientes afirmaciones acerca de una clase abstracta (*abstract class*) es más correcta:

- A) Una clase abstracta sólo puede definir métodos que sean abstractos, no puede tener variables miembro (campos).
- B) **No pueden crearse objetos de una clase abstracta.**
- C) Java permite la herencia múltiple únicamente si todas las super-clases son clases abstractas.
- D) Todas las afirmaciones anteriores son correctas.

Comentario

A) Es conveniente distinguir entre la “*declaración*” de un método (la primera línea o header del método) y su “*definición*” (todo el código del método, incluyendo la primera línea). Puede encontrarse una definición de estos dos conceptos, que son básicos en programación, en la página 9 del texto “Aprenda Java como si estuviera en primero”.

La respuesta A hace referencia a la *definición* de métodos.

En las clases abstractas pueden definirse métodos que no sean abstractos (ver páginas 235 y 236 del texto “Piensa en Java”, o la página 47 del texto “Aprenda Java como si estuviera en primero”). Asimismo, pueden definirse variables miembro (ver, por ejemplo, la clase abstracta Instrumento, en la página 236 del texto “Piensa en Java”). Por ello, la respuesta A es **falsa**.

B) **Verdadera**. Ver páginas 235 y 236 del texto “Piensa en Java”.

C) Java no permite la herencia múltiple de clases abstractas (ver página 259 del texto “Piensa en Java”, o las páginas 12, 35, 45 y 48 del texto “Aprenda Java como si estuviera en primero”). Por ello, la respuesta C es **falsa**.

D) **Falsa**, ya que la respuesta B es verdadera.

4.- Señale cual de las siguientes afirmaciones acerca de una *interface* es correcta.

A) Todos los métodos de una interface deben ser abstractos.

B) Una clase puede heredar varias interfaces, debiendo obligatoriamente definir todos los métodos declarados en las interfaces.

C) Las interfaces pueden definir variables finales (constantes).

D) **Todas las afirmaciones anteriores son correctas.**

Comentario

A) Puede encontrarse la definición del concepto de “método abstracto” en las páginas 235 y 255 del texto “Piensa en Java”: es aquel método del cual sólo se da su declaración, no su definición.

Todos los métodos de una interfaz deben ser abstractos, ya que no puede especificarse su definición (ver las páginas 255 y 256 del texto “Piensa en Java”, o la página 49 del texto “Aprenda Java como si estuviera en primero”). Por tanto, la respuesta A es **verdadera**.

B) **Verdadera** (ver páginas 258 a 260 del texto “Piensa en Java”, o las páginas 12 y 48 del texto “Aprenda Java como si estuviera en primero”).

C) La interface puede definir campos, que son implícitamente estáticos y constantes (ver la página 255 del texto “Piensa en Java” o la página 48 del texto “Aprenda Java como si estuviera en primero”). Por ello, la respuesta es **verdadera**.

D) **Esta es la respuesta correcta**, ya que las tres anteriores son correctas.

5.- ¿Qué es un constructor por defecto de una cierta clase?

A) **Un constructor con el mismo nombre que la clase y sin parámetros.**

B) Un constructor que devuelve la clase base.

C) Un constructor sobrecargado.

D) Un constructor que crea elementos de esa clase.

Comentario

Por la propia definición de constructor. Página. 136 Piensa en Java

6.- La clase ClaseB es sub-clase de ClaseA. En la clase ClaseA se define un único método, *metodoA*. En la clase ClaseB se define un único método, *metodoB*. En el siguiente fragmento de código, se pretende que una referencia a un objeto de la clase ClaseA sirva de referencia para un objeto de la clase ClaseB.

```
ClaseA a;  
a = new ClaseB( );
```

Indique, de las siguientes afirmaciones, cual es correcta:

A) Una referencia a un objeto de la clase ClaseA no puede servir de referencia para un objeto de la clase ClaseB. Por ello, el fragmento anterior de código es incorrecto.

B) **El objeto al que referencia a puede ser usado con el método *metodoA*, pero no con el método *metodoB*.**

C) El objeto al que referencia a puede ser usado con el método *metodoB*, pero no con el método *metodoA*.

D) El objeto al que referencia a puede ser usado con el método *metodoA* y con el método *metodoB*.

Comentario

Los conceptos de “*sub-clase*” (o “clase derivada”) y “*super-clase*” (o “clase base”) son básicos en la programación orientada a objetos, y por tanto el alumno debe conocerlos. Si ClaseA es *super-clase* de ClaseB, entonces ClaseB es *sub-clase* de ClaseA. Entre otros textos, puede encontrarse la definición y empleo de estos términos en “Aprenda Java como si estuviera en primero”, en las páginas 9, 16, 45 a 47, etc.

A) Una referencia a un objeto de una determinada clase es capaz de servir de referencia o de nombre a objetos de

cualquiera de sus sub-clases (es decir, de sus clases derivadas). Ver las páginas 227 a 230 del texto “Piensa en Java”, o la página 16 del texto “Aprenda Java como si estuviera en primero”. Por tanto, la respuesta A es **falsa**.

B) **Verdadera**. Ver las páginas 249 a 251 del texto “Piensa en Java”, o la página 16 del texto “Aprenda Java como si estuviera en primero”.

C) **Falsa**. Ver las páginas 249 a 251 del texto “Piensa en Java”, o la página 16 del texto “Aprenda Java como si estuviera en primero”.

D) **Falsa**. Ver las páginas 249 a 251 del texto “Piensa en Java”, o la página 16 del texto “Aprenda Java como si estuviera en primero”.

7.- Dado el siguiente código

```
public class prueba{
    int i =0;
    prueba incremento (){
        i++;
        return this;
    }
}
```

¿Qué valor devolverá la orden RETURN THIS?

A) **una referencia al objeto que usa la función incremento**

B) $i=1$

C) no devuelve nada

D) ninguna de las anteriores es correcta

Comentario

El operador `this` se utiliza comúnmente para pasar una referencia al objeto actual como parámetro para otros métodos. Por tanto la solución correcta es la a).

8.- Cuando se crea un elemento de una clase:

A) Se inicializa todo (variables y constructores) según el orden en el que está escrito

B) **Primero se inicializan las variables y luego los constructores**

C) Primero los constructores y luego las variables.

D) No existe ningún orden establecido.

Comentario

Página 156. Piensa en Java

9.- Indique si son correctas las siguientes dos afirmaciones:

I) Las funciones miembro (métodos) tienen acceso directo a todas las variables miembro (campos) de la clase, sin necesidad de anteponer el nombre de un objeto de la clase.

II) Las funciones miembro de una clase *B* derivada de otra *A*, tienen acceso a todas las variables miembro (campos) de *A* declaradas como *public* o *protected*, pero no a las declaradas como *private*.

A) **I: Sí; II: Sí**

B) I: Sí; II: No

C) I: No; II: Sí

D) I: No; II: No

Comentario

I) **Verdadera**. Ver la página 24 del texto “Aprenda Java como si estuviera en primero”.

II) **Verdadera**. Ver la página 179 del texto “Piensa en Java”, o la página 24 del texto “Aprenda Java como si estuviera en primero”.

La respuesta correcta es la A.

10.- Dado el siguiente código de un método cual será el resultado

```
void incremento () {
    int i;
```

```
        i++;  
    }
```

- A) **Producirá un error puesto que i no está inicializada**
- B) 3
- C) Un puntero a la variable i
- D) Ninguna de las anteriores es correcta

Comentario

La declaración de una variable en un método debe ir acompañada de su inicialización si no dará un error en la compilación. La solución es la a) .Evidentemente para dar un error será en la fase compilación si fuera en ejecución sería una excepción por tanto no vale la d.

11.- Considérese el siguiente fragmento de código:

```
ClaseA var1 = new ClaseA();  
ClaseA var2 = var1;
```

Indique cual de las siguientes respuestas es correcta:

- A) var1 y var2 son dos instancias de la clase ClaseA.
- B) var1 y var2 son el mismo objeto de la clase ClaseA.
- C) **var1 y var2 son dos referencias a un mismo objeto de la clase ClaseA.**
- D) var1 es un objeto de la clase ClaseA. var2 es una referencia al objeto var1.

Comentario

Respuesta correcta: C. Ver las páginas 80-81 y 127-129 del texto “Piensa en Java”, o la página 6 del texto “Aprenda Java como si estuviera en primero”.

12.-Supóngase que se tiene definida una clase denominada perro. ¿Qué efecto produce la siguiente sentencia?

```
new perro();
```

- A) Inicializa un elemento de la clase perro.
- B) **Llama al constructor por defecto de la clase perro.**
- C) Reserva espacio de memoria para un objeto de la clase perro y lo inicializa.
- D) Ninguna de las anteriores.

Comentario

Solución Libro Piensa en Java página 137

13.-Dentro de una unidad de compilación:

- A) Debe haber una clase pública
- B) Pueden existir varias clases públicas
- C) **Puede haber una única clase pública**
- D) Ninguna de las anteriores

Comentario

Sol. Libro Piensa en Java página 170

14.- Al llamar a un método sobrecargado, Java sigue unas reglas para determinar el método concreto que debe llamar. Indique si Java aplica cada una de las siguientes dos reglas:

- I) Si no existe un método que se ajuste exactamente al tipo de los argumentos de la llamada, siempre se produce un error.
- II) El valor de retorno influye en la elección del método sobrecargado. Por ello, es posible crear dos métodos sobrecargados, es decir con el mismo nombre, que sólo difieran en el valor de retorno.

A) I: Sí; II: Sí

B) I: Sí; II: No

C) I: No; II: Sí

D) I: No; II: No

Comentario

I) **Falsa.** Si no existe un método que se ajuste exactamente al tipo de los argumentos de llamada, se intenta promover los argumentos actuales al tipo inmediatamente superior (por ejemplo, *char* a *int*, *int* a *long*, *float* a *double*, etc.) y se llama al método correspondiente. Ver las páginas 132-134 del texto “Piensa en Java”, o la página 40 del texto “Aprenda Java como si estuviera en primero”.

II) **Falsa.** El valor de retorno no influye en la elección del método sobrecargado. Ver la página 136 del texto “Piensa en Java”, o la página 40 del texto “Aprenda Java como si estuviera en primero”.

La respuesta correcta es la D.

15.-¿Cuál será la salida del siguiente fragmento de código?

```
class perro{
    perro(int i) {System.out.println(i);
    }
public class mastin extends perro{
    }
public static void main(String[] args){
    mastin chucho=new mastin(1);
    }
}
```

- A) null
- B) 1
- C) 0
- D) **Ninguna, el código es erróneo**

Comentario

Debido a que la clase base no presenta un constructor por defecto, es necesario mediante la sentencia `super(i)` invocar antes a la clase base

16. ¿Qué método de la interfaz `AppletContext` recibe como argumento un objeto `URL` y muestra en una ventana de un navegador el recurso asociado con la `URL` que se le indica.

- A) **showDocument**
- B) showURL
- C) getDocument
- C) getURL

Comentario

La respuesta correcta es la a). Los métodos de las respuestas b), c) y d) realizan otras funciones. Consultar las páginas 923 y 924 del libro “Thinking in Java, 2th. ed.”.

17. Cuando se crea un `ServerSocket` el parámetro que necesita el constructor es:

- A) **Un número de puerto.**
- B) Un nombre de una máquina.
- C) Una dirección IP.
- D) Una URL.

Comentario

La respuesta correcta es la a) ya que el parámetro que siempre se necesita para que un servidor permanezca a la escucha es el número de puerto en el que atender a los clientes. Se explica más detalladamente y con ejemplos en la página 909 del libro “Thinking in Java, 2th. ed.” y en la documentación de Sun.

18. ¿Por qué un programa que realiza labores de servidor extiende la clase `Thread`?

- A) **Para poder atender varias conexiones simultáneamente.**
- B) Para poder trabajar en un sistema operativo multihilo.
- C) Es una condición necesaria para poder utilizar la clase `Socket`.
- D) Para poder atender varias conexiones de forma secuencial.

Comentario

La respuesta correcta es la a) ya que para que un servidor pueda atender varias conexiones simultáneamente necesita tener tantos procesos trabajando en paralelo como conexiones existen en un instante dado. Ello se consigue mediante la creación de threads. Se explica más detalladamente y con ejemplos en la página 917 del libro “Thinking in Java, 2th. ed.”.

19. La clase Java que se utiliza para representar una dirección IP es

- A) **InetAddress.**
- B) IPAddress.
- C) NumberIPAddress.
- D) InternetAddress.

Comentario

La a) es la única respuesta posible ya que las otras tres no existen. Página 905 del libro "Thinking in Java, 2th. ed."

20. La palabra reservada *synchronized* se utiliza para:

- A) **Evitar los accesos simultáneos a un recurso.**
- B) Permitir los accesos simultáneos a un recurso.
- C) Sincronizar las llamadas al método con la prioridad del thread.
- D) Evitar que un método se quede bloqueado al acceder a un recurso compartido.

Comentario

*Sólo un thread en un instante dado puede llamar a un método *synchronized* para acceder a un objeto concreto. La respuesta correcta es la a) que se explica en las páginas 59 y 848 del libro "Thinking in Java, 2th. ed."*