

Pregunta 1.

☐

Puntos: 1

1. Verdadero/Falso: Diseño modular - Pregunta 1: En un programa Java es obligatorio cr...

▼

Pregunta	En un programa Java es obligatorio crear una función y un procedimiento.
Respuesta	Verdadero ✓ Falso
Comentario correcto	Muy bien.
Comentario incorrecto	No. No es obligatorio. Puede ser recomendable pero no obligatorio.

Pregunta 2.

☐

Puntos: 1

2. Varias respuestas correctas: Diseño modular - Pregunta 2: ¿Cuál de las siguientes definiciones ...

▼

Pregunta	¿Cuál de las siguientes definiciones de funciones/procedimientos es incorrecta en Java?
Respuesta	<div>public static void nombre()</div> <div>public static int procedimiento(String entero, int cadena)</div> <div>✓ public static miFuncion(String cadena)</div> <div>✓ public static void miFuncion(int numero1, numero2)</div>
Comentario correcto	Correcto.
Comentario incorrecto	1.- public static miFuncion(String cadena) Falta indicar el tipo de dato que retorna la función o void si no retorna nada. 2.- public static void miFuncion(int numero1, numero2) Falta indicar el tipo de dato para el segundo parámetro que recibe la función

Pregunta 3.

Puntos: 1



3. Varias respuestas correctas: Diseño modular - Pregunta 3: Indica la respuesta correcta: Una fun...



Pregunta	Indica la respuesta correcta: Una función o procedimiento se compone de:
Respuesta	<div>Una entrada y una salida</div> <div>Una entrada o entradas determinadas</div> <div>Varios datos de salida</div> <div>Un nombre único e identificable en el algoritmo del programa principal</div> <div> Un dato o datos de entrada, un dato de salida y un nombre único.</div> <div>Un algoritmo que hace algo a partir de unos datos previamente establecidos en la función.</div>
Comentario correcto	Correcto
Comentario incorrecto	Una función o procedimiento se compone de: - Una entrada o varias - Un dato de salida que puede ser vacío (void) - Un nombre único e identificable.

Pregunta 4.

☐

Puntos: 1

4. Verdadero/Falso: Diseño modular - Pregunta 4: Una función solo puede ser llamada de...

▼

Pregunta	Una función solo puede ser llamada desde la función de arranque: public static void main(String [] args)
Respuesta	<div>Verdadero</div> <div>✓ Falso</div>
Comentario correcto	Muy bien
Comentario incorrecto	No. Una función puede ser llamada desde el método main (arranque) o desde otra función/procedimiento.

Pregunta 5.

☐

Puntos: 1

5. Varias respuestas correctas: Diseño modular - Pregunta 5: Indica las respuestas correctas:

▼

Pregunta	Indica las respuestas correctas:
Respuesta	<div>Una variable definida en el método de arranque: public static void main(String [] args) puede ser utilizada en dicho método y no puede ser pasada a ninguna función definida en el mismo fichero .java</div> <div>✓ Una variable definida en una función o procedimiento solo puede ser utilizada dentro de esa función o procedimiento</div> <div>✓ Un argumento o parámetro es un dato de entrada en una función o procedimiento.</div> <div>Un parámetro o argumento de una función puede ser utilizado fuera de ella.</div>
Comentario incorrecto	<div>Una variable definida en el método de arranque: public static void main(String [] args) puede ser utilizada en dicho método y no puede ser pasada a ninguna función que es llamada desde dicho método y definida en el mismo fichero .java</div> <div>Incorrecto: puede pasarse a una función o procedimiento que es llamado desde la función de arranque main.</div> <div>Un parámetro o argumento de una función puede ser utilizado fuera de ella.</div> <div>Incorrecto: Un parámetro o argumento de una función no puede ser utilizado fuera de ella.</div>

Pregunta 6.



Puntos: 1

6. Verdadero/Falso: Diseño modular - Pregunta 6: Una librería es un conjunto de utilid...



Pregunta	Una librería es un conjunto de utilidades ya implementadas que facilitan la tarea del programador.
Respuesta	<input checked="" type="checkbox"/> Verdadero <input type="checkbox"/> Falso
Comentario correcto	Correcto.
Comentario incorrecto	La definición de una librería es esta: "Una librería se puede entender como un conjunto de clases, funciones y procedimientos que facilitan la tarea ofreciendo al programador funcionalidad que ya está codificada y lista para ser usada."

Pregunta 7.



Puntos: 1

7. Varias respuestas correctas: Diseño modular - Pregunta 7: Indica las respuestas incorrectas:



Pregunta	Indica las respuestas incorrectas:
Respuesta	<input checked="" type="checkbox"/> Una librería permite duplicar las mismas líneas de código. <input checked="" type="checkbox"/> Solo puede ser definida dentro del método de arranque. <input type="checkbox"/> Permite la reutilización y por tanto: minimiza los errores y disminuye el tiempo de desarrollo. <input type="checkbox"/> En Java una librería tiene extensión .jar
Comentario correcto	Correcto.
Comentario incorrecto	Una librería permite duplicar las mismas líneas de código. Incorrecto. Es justo al contrario. Evitamos duplicidades en el código. Solo puede ser definida dentro del método de arranque. Incorrecto. Debe ser definida fuera del método de arranque.

Pregunta 8.



Puntos: 1

8. Varias respuestas correctas: Diseño modular - Pregunta 8: Indica si las implementaciones de las...



Pregunta	Indica si las implementaciones de las funciones/procedimientos descritos son correctas y no producen ningún tipo de error:
Respuesta	<div><pre>public static void suma(int num1, int num2){ return num1+num2; }</pre></div> <div><pre>public static int miFuncion(){ int num; return num1+10; }</pre></div> <div><div>✔</div><pre>public static void miProcedimiento(int numero){ }</pre></div> <div><div>✔</div><pre>public static int miProcedimiento(String numero, int inte){ return 10; }</pre></div>
Comentario correcto	Muy bien!

Comentario
incorrecto

```
public static void suma(int num1, int num2){  
    return num1+num2;  
}
```

Error: Si en la cabecera de la función hay un "void" no se puede utilizar la palabra reservada "return".

```
public static int miFuncion(){  
    int num;  
    return num1+10;  
}
```

Error: La variable num1 no está definida en la función de ninguna manera.


Pregunta 9.



Puntos:

9. Verdadero/Falso: Diseño modular - Pregunta 9: La recursividad es la capacidad de un...



Pregunta	La recursividad es la capacidad de una función de llamarse a sí misma, repitiendo el proceso hasta que valga 1, siendo esta la condición de salida para finalizar las llamadas.
Respuesta	<div>Verdadero</div> <div> Falso</div>
Comentario correcto	Correcto.
Comentario incorrecto	Incorrecto. La condición de salida no tiene porqué ser 1. Este es el caso del cálculo del factorial, pero para otra función recursiva puede ser otra condición diferente.

Pregunta 10.



Puntos:

10. Numérica calculada: Diseño modular - Pregunta 10: Dada la siguiente función indica su r...



Pregunta	<div>Dada la siguiente función indica su resultado.</div> <div><pre>public static int calculaFormula(){ int num = 10; int tmp = 5; String cadena = "3"; int temporal=1; temporal = Integer.parseInt(cadena)+temporal; tmp += num+temporal; return 10+5 * temporal+tmp; }</pre></div>
Respuesta	49
Rango de respuesta +/-	0
Comentario correcto	Muy bien.

Comentario
incorrecto

```
public static int calculaFormula(){  
    int num = 10;  
    int tmp = 5;  
    String cadena = "3";  
    int temporal=1;  
  
    temporal = Integer.parseInt(cadena)+temporal;  
    //Tras la instrucción anterior temporal vale: 3+1 = 4  
    tmp += num+temporal;  
    //Tras la instrucción anterior tmp vale 10+(5+4)=19  
    return 10+5 * temporal+tmp;  
    //Se retorna 10+5*4+19. Por el orden de prevalencia, primero se ejecuta  
    //la multiplicación y luego las sumas.  
    //Luego: 10 + (5*4) + 19 = 49  
    //Se retorna 49  
}
```