UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA – ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
71901072 – PROGRAMACIÓN ORIENTADA A OBJETOS (GRADO EN INGENIERÍA INFORMÁTICA / TECNOLOGÍAS DE LA INFORMACIÓN)

PROBLEMAS RESUELTOS

V.1. CURSO 2013-14

A continuación el equipo docente presenta un conjunto de problemas tipo test que pueden servir para ayudar al alumno a prepararse para el examen. Son problemas que han sido extraídos de exámenes anteriores y/o están disponibles libremente en diferentes colecciones de problemas de este tipo en Internet. La ventaja que ofrece este documento en comparación a las otras fuentes de problemas es que éstos están organizados por temas, así que un alumno puede terminar de estudiar un tema y a continuación aplicar sus conocimientos con los problemas del tema correspondiente. Hay dos versiones disponibles del documento: una versión con las respuestas marcadas y otra sin ellas.

CAPÍTULO 1. OBJETOS Y CLASES

Pregunta: Dado el siguiente fragmento de código, cuál es el resultado del comando **java test 2**:

```
public class test {
   public static void main(String args[]) {
      Integer intObj=Integer.valueOf(args[args.length-1]);
      int i = intObj.intValue();

      if(args.length > 1)
         System.out.println(i);
      if(args.length > 0)
         System.out.println(i - 1);
      else
         System.out.println(i - 2);
   }
}
```

a. test
b. test-1
c. 1
d. 2

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es **incorrecta**:

a. Las clases definen tipos.
b. Las diagramas de clases muestran las clases de una aplicación y la relación entre ellos.
c. Las clases son objetos.
d. Las clases definen métodos.

Pregunta: ¿Cuál de las siguientes es una **palabra reservada** en Java?:

a. NULL
b. new
c. instanceOf
d. wend

Pregunta: ¿Cuál de las siguientes instrucciones compila sin provocar un warning o un error?

a. `char c=”a”;`
b. `byte b=257;`
c. `boolean b=null;`
d. `int i=10;`

Pregunta: ¿Cuál de los siguientes no es un identificador válido en Java?

a. #variable
b. $variable
c. _variable
d. vari_able

Pregunta: Dada la declaración de las siguientes variables, indicar cuáles de ellas son correctas.

```
1.      float foo = -1;
2.      float foo1 = 1.0;
3.      float foo2 = 42e1;
4.      float foo3 = 2.02f;
5.      float foo4 = 3.03d;
6.      float foo5 = 0x0123;
```

a. 1 y 2
b. 1 y 3
c. 4 y 6
d. 3 y 4

Pregunta: Respecto a los bucles, indique cuál de las siguientes afirmaciones es falsa:

a. El cuerpo de un bucle for-each puede repetirse 0 o más veces.
b. Un bucle for-each puede aplicarse sobre cualquier clase que implemente la interfaz Iterable.
c. El cuerpo de un bucle while siempre se ejecuta, como mínimo, una vez.
d. Un bucle for-each puede aplicarse sobre arreglos (arrays).

Pregunta: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
public class Test
{
     public static void main(String args[])
     {
          char c = -1;
          System.out.println(c);
     }
}
```

a. La expresión "`char c = -1;`" provocará un error de compilación debido a que el rango de la clase "`char`" es $0-2^{(16-1)}$.
b. No habrá error de compilación, la salida será -1.
c. No habrá error de compilación, la salida no será ningún carácter ascii.
d. No habrá error de compilación, la salida será un carácter Unicode.

Pregunta: Según la bibliografía básica, ¿Qué elementos cree que definen a un objeto?

a. Sus cardinalidad y su tipo
b. Sus atributos y sus métodos
c. La forma en que establece comunicación e intercambia mensajes
d. Su interfaz y los eventos asociados

Pregunta: De acuerdo a la bibliografía básica ¿Qué significa instanciar una clase?

a. Duplicar una clase
b. Eliminar una clase
c. Crear un objeto a partir de la clase
d. Conectar dos clases entre sí

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. La signatura es el encabezado de un método y proporciona la información necesaria para invocarlo.
b. La signatura está formada por los parámetros de un método y proporciona la información necesaria para invocarlo.
c. La signatura es el nombre de un método y puede tener parámetros para proporcionar información adicional para realizar una tarea.
d. La signatura es el encabezado de un método y puede tener parámetros para proporcionar información adicional para realizar una tarea.

Pregunta: ¿Cuál de las siguientes sentencias son correctas?

```
(1) int w = (int)888.8;
(2) byte x = (byte)1000L;
(3) long y = (byte) 100;
(4) byte z = (byte) 100L;
```

a. 1 y 2.
b. 2 y 3.
c. 3 y 4.
d. Todas son correctas.

CAPÍTULO 2. COMPRENDER LAS DEFINICIONES DE CLASES

Pregunta: ¿Cuál es el resultado de ejecutar el siguiente código?

```
public class Ejemplo
{
      private int i= dameJ();
      private int j=10;

      private int dameJ ()
      {
            return j;
      }

      public static void main (String []args)
      {
            System.out.println ((new Ejemplo()).i);
      }
}
```

a. Da un error de compilación debido a las restricciones de acceso a las variables privadas de Ejemplo.
b. Da un error de compilación debido a la referencia que se hace a métodos declarados con posterioridad.
c. No da ningún error de compilación y produce como salida el valor 0.
d. No da ningún error de compilación y produce como salida el valor 10.

Pregunta: ¿Cuál es el resultado de ejecutar el siguiente código?

```
public class Ejemplo
{
      private int i=j;
      private int j=10;
      public static void main (String []args)
      {
```

```
            System.out.println ((new Ejemplo()).i);
        }
    }
```

a. Da un error de compilación debido a las restricciones de acceso a las variables privadas de Ejemplo.
b. Da un error de compilación debido a la referencia que se hace a variables declaradas con posterioridad.
c. No da ningún error de compilación y produce como salida el valor 0.
d. No da ningún error de compilación y produce como salida el valor 10.

Pregunta: Dado el siguiente fragmento de código, indica cuál de las siguientes afirmaciones es correcta en relación al valor de la variable `foo`.

Número de Línea	Código
4	`int index = 1;`
5	`boolean[] test = new boolean[3];`
6	`boolean foo = test [index];`

a. `foo` tiene el valor 0
b. `foo` tiene el valor `null`
c. `foo` tiene el valor `false`
d. Se produce una excepción y `foo` no posee ningún valor

Pregunta: Indique el resultado de ejecutar el siguiente código que se muestra a continuación:

Número de Línea	Código
4	`public class test {`
5	`public static void add3 (Integer i) {`
6	`int val = i.intValue();`
7	`val += 3;`
8	`i = new Integer (val);`
9	`}`
10	`public static void main (String args[]) {`
11	`Integer i = new Integer (0);`
12	`add3 (i);`
13	`System.out.println (i.intValue ( ) );`
14	`}`
15	`}`

a. El programa indicará un fallo en tiempo de compilación.
b. El programa imprime por pantalla el valor “0”.
c. El programa imprime por pantalla el valor “3”.
d. El programa lanzará una excepción en la línea 6 (`int val = i.intValue();`).

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Los métodos pueden devolver información de algún objeto mediante un valor de retorno.
b. Los métodos siempre tienen parámetros con los que obtener la información necesaria.
c. A partir de una clase tan solo se puede crear un solo objeto.
d. El estado de los objetos se representa mediante los parámetros de su constructor.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Los campos se conocen como variables de objeto.

b. El alcance de una variable define la sección de código desde donde la variable puede ser declarada.
c. Los constructores permiten que cada objeto sea preparado adecuadamente cuando es creado.
d. El tiempo de vida de una variable describe el número de veces que es utilizada en un método.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuales de las siguientes expresiones resultan verdaderas:

```
1. ! ( 4 < 5 )
2. ( 2 > 2 ) || ( ( 4 == 4 ) && ( 1 < 0 ) )
3. ( 2 > 2 ) || ( 4 == 4 ) && ( 1 < 0 )
4. ( 2 > 2 ) || !( ( 4 == 4 ) && ( 1 < 0 ) )
5. ( 34 != 33 ) && ! false
```

a. Las expresiones 3 y 4.
b. Las expresiones 2 y 4.
c. Las expresiones 3 y 5.
d. Las expresiones 4 y 5.

Pregunta: Dadas las siguientes expresiones, indica cuál de las opciones es la correcta.

1. `(1 > 1) && (1 > 1) == (1 > 1) == false`
2. `(1 == 1) | (10 > 1) == true | true == true`

a. La expresión 1 es evaluada como falsa y la expresión 2 como falsa.
b. La expresión 1 es evaluada como falsa y la expresión 2 como verdadera.
c. La expresión 1 es evaluada como verdadera y la expresión 2 como falsa.
d. La expresión 1 es evaluada como verdadera y la expresión 2 como verdadera.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Los métodos pueden devolver información de algún objeto mediante un valor de retorno.
b. Los métodos siempre tienen parámetros con los que obtener la información necesaria.
c. A partir de una clase tan solo se puede crear un solo objeto.
d. El estado de los objetos se representa mediante los parámetros de su constructor

Pregunta: Según el texto de la bibliografía básica de la asignatura, el alcance de una variable …

a. Define la forma en la que la variable puede ser accedida.
b. Define el conjunto de métodos que puede acceder a la variable.
c. Define la sección de código en la que la variable puede ser accedida.
d. Ninguna de las anteriores.

Pregunta: De acuerdo a la bibliografía básica, ¿Cuál es la descripción que crees que define mejor el concepto 'clase' en la programación orientada a objetos?

a. Es un concepto similar al de 'array'
b. Es un tipo particular de variable
c. Es un modelo o plantilla a partir de la cual creamos objetos
d. Es una categoría de datos ordenada secuencialmente

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Los campos se conocen como variables de objeto.
b. El alcance de una variable define la sección de código desde donde la variable puede ser declarada.
c. Los constructores permiten que cada objeto sea preparado adecuadamente cuando es creado.
d. El tiempo de vida de una variable describe el número de veces que es utilizada en un método.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Los campos se definen dentro de los constructores y de los métodos.
b. Los campos se usan para almacenar datos que nunca persisten durante la vida del objeto.
c. Los campos tienen un tiempo de vida que perdura después de terminar el objeto.
d. La accesibilidad de los campos se extiende a toda clase y por este motivo pueden usarse dentro de cualquier constructor o método de clase en la que estén definidos.

Pregunta: Dado el siguiente código, el resultado será:

```
class MiClase {public int valor;}

class Test {

      public static void main(String[] args){
      MiClase a1 = new MiClase ();
      MiClase a2 = new MiClase ();
      MiClase a3 = new MiClase ();
      a1.valor=150;
      a2.valor=150;
      a3 = a2;
      if (a1 == a2) { System.out.println(" UNO");}
      if (a1 == a3) { System.out.println(" DOS");}
      if (a2 == a3) { System.out.println(" TRES");}

}

}
```

a. UNO
b. UNO TRES
c. UNO DOS TRES
d. TRES

Pregunta: Dado el siguiente fragmento de código, indica cuál de las siguientes afirmaciones es correcta en relación al valor de la variable `foo`.

Número de Línea	**Código**
4	`int index = 1;`
5	`boolean[] test = new boolean[3];`
6	`boolean foo = test [index];`

a. `foo` tiene el valor 0
b. `foo` tiene el valor `null`
c. `foo` tiene el valor `false`
d. Se produce una excepción y `foo` no posee ningún valor

CAPÍTULO 3. INTERACCIÓN DE OBJETOS

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. La depuración es la actividad cuyo objetivo es determinar si una pieza de código produce el comportamiento pretendido.
b. La prueba viene a continuación de la depuración.

c. La depuración es una actividad dedicada a determinar si un segmento de código contiene errores.
d. La depuración es el intento de apuntar con precisión y corregir un error en el código.

Pregunta: Indique cuál de las siguientes afirmaciones es falsa:

a. El objetivo de la sobrecarga de métodos es facilitar la invocación de un mismo método pasándole un conjunto de parámetros de entrada diferentes.
b. Se puede sobrecargar un método variando el tipo de retorno de éste sin variar los parámetros de entrada.
c. Un método puede ser sobrecargado en la misma clase o en una subclase.
d. Los métodos sobrecargados pueden cambiar el modificador de acceso del método original.

Pregunta: Sea la siguiente definición de clase:

```
public class ClaseA {
     public ClaseA(String s) { System.out.print("Construyendo Clase A."); }
}
```

Y la siguiente definición de una subclase:

```
public class ClaseB extends ClaseA {

     public ClaseB(String s) { System.out.print("Construyendo Clase
B.");super(s); }

     public static void main(String [] args) {
          new ClaseB("Objeto Clase B");
          System.out.println(" ");
     }
}
```

¿Cuál de las siguientes afirmaciones es cierta al ejecutar el código?

a. Se produce un error en tiempo de ejecución.
b. Se muestra el mensaje “Construyendo Clase B. Construyendo Clase A.”.
c. Se muestra el mensaje “Construyendo Clase A. Construyendo Clase B”.
d. Error de compilación.

Pregunta: Dados las siguientes definiciones de clases:

```
public abstract class Disparo {
   protected int velocidad=10;
   abstract public void disparar();
}

public class DisparoUFO extends Disparo {
   public void disparar() {
        this.velocidad=20;
        System.out.println("Dispara la nave");
   }
}

public class DisparoNave extends Disparo{
   public void disparar() {
        this.velocidad=10;
        System.out.println("Dispara la nave");
   }
}
```

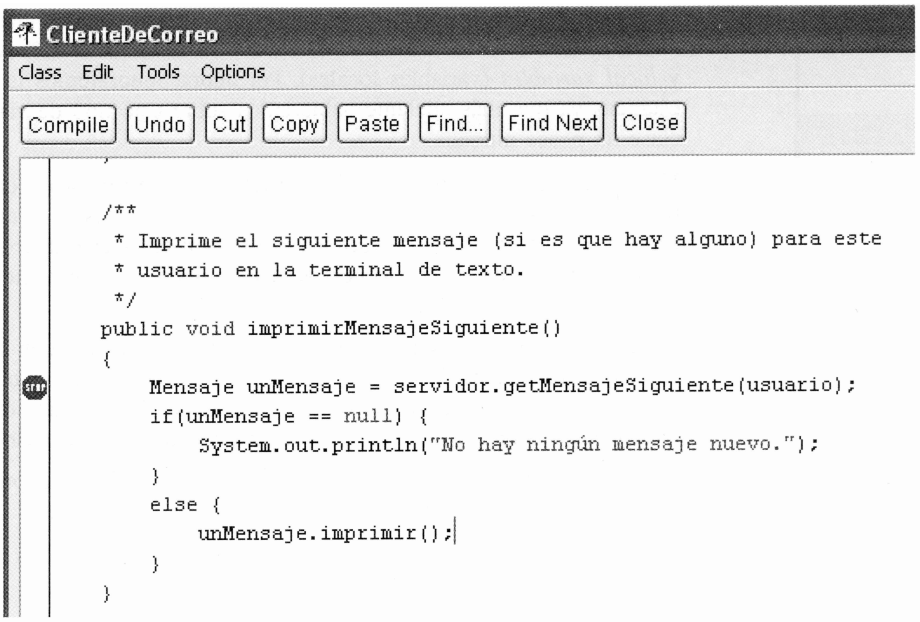
```
public class TestUFO {
   public static void main(String[] args) {
         Disparo dn=new DisparoNave();
         new TestUFO().inicio(dn);
   }
   public void inicio(Disparo d)
   {
         d.disparar();
   }
}
```

Podemos afirmar:

a. El método disparar está sobrecargado.
b. Muestra por pantalla el mensaje "Dispara la nave".
c. No se muestra por pantalla ningún mensaje.
d. Obtenemos un error en tiempo de ejecución.

Pregunta: La siguiente figura muestra una captura de pantalla del editor BlueJ con una línea de código recuadrada. Indica cuál de las siguientes afirmaciones es correcta en relación a la línea recuadrada:

a. Muestra un error en tiempo de ejecución.
b. Muestra un error de compilación.
c. Muestra un punto de interrupción.
d. Muestra una el lanzamiento de una excepción.



Pregunta: Dada la siguiente definición de clase, ¿Cuál sería el contenido más coherente a implementar en el constructor?

```
class Test {
      int var;
      Test(int var) { CONTENIDO CONSTRUCTOR }
}
```

a. var=var;
b. int var=var;
c. this.var=var;
d. No se puede llamar igual el parámetro del constructor que el atributo de la clase

Pregunta: De acuerdo a la bibliografía básica ¿Qué significa sobrecargar un método?

a. Editarlo para modificar su comportamiento
b. Cambiarle el nombre dejándolo con la misma funcionalidad
c. Crear un método con el mismo nombre pero diferentes argumentos
d. Añadirle funcionalidades a un método

Pregunta: Dado el siguiente fragmento de código que pretende mostrar un ejemplo de sobrescritura:

```
class Examen {
     private float pregunta = 1.0f ;
     protected float getNota () {return pregunta;}
}

class Test extends Examen {
     private float nota = 2.0f;
     //Insertar código aquí

}
```

Indique cual de las siguientes opciones completaría el código anterior para dar lugar a un ejemplo correcto de sobrescritura:

a. public float getNota (float valor) { return valor;}
b. public float getNota ( ) { return nota;}
c. float getNota ( ) { return nota;}
d. float double getNota ( ) { return nota;}

Pregunta: Dadas las siguientes expresiones, indica cuál de las opciones es la correcta.

1. `(1 > 1) && (1 > 1) == (1 > 1) == false`
2. `(1 == 1) | (10 > 1) == true | true == true`

a. La expresión 1 es evaluada como falsa y la expresión 2 como falsa.
b. La expresión 1 es evaluada como falsa y la expresión 2 como verdadera.
c. La expresión 1 es evaluada como verdadera y la expresión 2 como falsa.
d. La expresión 1 es evaluada como verdadera y la expresión 2 como verdadera.

Pregunta: ¿Cual es la salida del siguiente código?

```
5. int x = 5 * 4 % 3;
6. System.out.println(x);
```

a. Error de compilacion en la línea 5.
b. 2
c. 3
d. 6

Pregunta: ¿Cual es la salida del siguiente código?

```
3. int x = 10, y = 3;
4. if(x % y == 2)
5. System.out.print("dos");
6. System.out.print(x%y);
7. if(x%y == 1)
8. System.out.print("uno");
```

a. dos1
b. dos2

c. uno
d. 1uno

CAPÍTULO 4. AGRUPACIÓN DE OBJETOS

Pregunta: Dado el siguiente código, ¿cuál es su resultado?

Número de Línea	Código
4	`class Top {`
5	`public Top(String s) { System.out.print("B"); }`
6	`}`
7	`public class Bottom2 extends Top {`
8	`public Bottom2(String s) { System.out.print("D"); }`
9	`public static void main(String [] args) {`
10	`Bottom2 obj=new Bottom2("C");`
11	`System.out.println(" ");`
12	`}`
13	`}`

a. BD
b. DB
c. BDC
d. Error de compilación

Pregunta: Dada la siguiente clase Prueba:

```
public class Prueba {
       public static void main(String[] args) {
              ArrayList < Integer > valores = new ArrayList < Integer > ();
              valores.add(4);
              valores.add(5);
              valores.set(1, 6);
              valores.remove(0);
              for(Integer v : valores) {
                     System.out.print(v);
              }
       }

}
```

Al ejecutar el código obtendremos:

a. Un error en tiempo de ejecución
b. Se mostrará 4
c. Se mostrará 5
d. Se mostrará 6

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. El lenguaje Java tiene tres variantes del ciclo for : for-each, for y for-do.
b. Un ciclo while es similar en su estructura y propósito que el ciclo for-each.
c. El tipo de la variable de ciclo no tiene porqué ser el mismo que el tipo del elemento declarado para la colección que estamos recorriendo con un ciclo.
d. Un índice es un objeto que proporciona funcionalidad para recorrer todos los elementos de una colección.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Las colecciones de objetos son objetos que pueden almacenar un número predeterminado e invariable de otros objetos.

b. Un iterador es un objeto que proporciona funcionalidad para recorrer todos los elementos de una colección.
c. Un ciclo consiste en la escritura repetida de un bloque de sentencias.
d. Un arreglo (array) es un tipo especial de colección que puede almacenar un número variable de elementos.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta en relación a la clase Vector de Java:

a. Es Final
b. Implementa java.util.List
c. Es serializable
d. Dispone de un solo constructor

Pregunta: ¿Cuál de las siguientes sentencias declara legalmente, construye e inicializa un array?

```
a. int [] miLista = {"1", "2", "3"};
b. int [] miLista = (5, 8, 2);
c. int miLista [] [] = {4,9,7,0};
d. int miLista [] = {4, 3, 7};
```

Pregunta: Dado el siguiente fragmento de programa, indique que afirmación es cierta:

```
int cont;
for (cont=5 ; cont>0 ; cont--)
System.out.print(cont);
System.out.print(cont);
```

a. Se imprime en pantalla 543210
b. Se imprime en pantalla 5432100
c. Se imprime en pantalla 554433221100
d. Se imprime en pantalla 543210-1

Pregunta: Dado el siguiente fragmento de código:

```
int indice = 1;
boolean[] examen = new boolean[8];
boolean poo = examen [indice];
```

Indica cuál de las siguientes afirmaciones es correcta en relación al valor de la variable poo.

a. poo tiene el valor 0
b. poo tiene el valor null
c. poo tiene el valor false
d. Se produce una excepción y poo no posee ningún valor

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes afirmaciones es correcta:

a. El término acoplamiento describe cuánto se ajusta una unidad de código a una tarea lógica o a una entidad
b. El acoplamiento describe la conectividad de los propios objetos de una clase
c. Un encapsulamiento apropiado en las clases reduce el acoplamiento
d. Un sistema debilmente acoplado se caracteriza por la imposibilidad de modificar una de sus clases sin tener que realizar cambios en ninguna otra

Pregunta: ¿Cuál es la salida del siguiente programa?

```
for (int i = 0; i <= 4; i += 2) {
    System.out.print(i + " ");
}
System.out.println(i);
```

a. 0 2 4
b. 0 2 4 5
c. 0 1 2 3 4
d. La compilación falla.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. El lenguajes Java tiene tres variantes del ciclo for: for-each, for y for-do
b. Un ciclo while es similar en su estructura y propósito que el ciclo for-each
c. El tipo de la variable de ciclo no tiene porqué ser el mismo que el tipo del elemento declarado para la colección que estamos recorriendo con un ciclo
d. Un índice es un objeto que proporciona funcionalidad para recorrer todos los elementos de una colección

Pregunta: Dada la siguiente instrucción:

```
x = y--;
```

¿Cuál de las siguientes afirmaciones es verdadera DESPUÉS de ejecutarse la instrucción?

a. La instrucción da un error de compilación
b. x > y
c. x == y
d. x < y

Pregunta: ¿De qué clase deriva la clase ArrayList?

a. AbstractList
b. AbstractCollection
c. ArrayCollection
d. ListCollection

Pregunta: ¿Qué ocurre si se compila y ejecuta el siguiente código?

```
public class Q {
      public static void main(String argv[]){
            int anar[]=new int[5];
            System.out.println(anar[0]);
       }
}
```

a. Error: anar se referencia antes de ser inicializado
b. `null`
c. 0
d. 5

Pregunta: Dado el siguiente código, indique qué ocurriría al llamar al método `ejemplo();`

Número de Línea	Código
4	`class Examen {`
5	`  private int i;`
6	`  public void ejemplo () {`
7	`  for (int i=0; i<5; i++)`
8	`  System.out.print (this.i++);`
9	`  }`
10	`}`

a. Imprime 00000.
b. Imprime 01234.
c. Imprime infinitos ceros.
d. Se producirá un error en tiempo de compilación por no estar inicializada la propiedad i.

CAPÍTULO 5. COMPORTAMIENTOS MÁS SOFISTICADOS

Pregunta: Dado el siguiente código, ¿cuál de las afirmaciones es cierta?

Número de Línea	Código
4	`class Hotel {`
5	`  public int reservas;`
6	`  public void reservar() {`
7	`  reservas++;`
8	`  }`
9	`}`
10	`public class SuperHotel extends Hotel {`
11	`  public void reservar() {`
12	`  reservas--;`
13	`  }`
14	`  public void reservar(int size) {`
15	`  reservar();`
16	`  super.reservar();`
17	`  reservas += size;`
18	`  }`
19	`  public static void main(String[] args) {`
20	`  SuperHotel hotel = new SuperHotel();`
21	`  hotel.reservar(2);`
22	`  System.out.print(hotel.reservas);`
23	`  }`
24	`}`

a. Error de compilación
b. Lanza una excepción en tiempo de ejecución
c. 0
d. 2

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes afirmaciones es falsa:

a. Únicamente las clases que implementan la interfaz List permiten el uso de iteradores.
b. Un iterador es un objeto que proporciona funcionalidad para recorrer todos los elementos de una colección.
c. Un iterador permite recorrer cualquier tipo de colección hacia adelante utilizando el método next() combinado con el método hasNext() para comprobar si se ha alcanzado el final de la colección.
d. Una colección puede recorrerse tanto con un iterador como con un ciclo for-each. Ambas formas son equivalentes.

Pregunta: Indique cuál de las siguientes afirmaciones es verdadera:

a. Para definir una variable de instancia es necesario utilizar la palabra reservada **static**.
b. Un método estático puede acceder a cualquier componente (método o variable) no estático de su clase.
c. Los métodos estáticos pueden ser sobreescritos.
d. Una variable de clase puede ser modificada sin necesidad de haber instanciado objeto alguno de dicha clase.

Pregunta: Dado el siguiente código, ¿Cuál de las siguientes afirmaciones es correcta?

```
Set < Object > objetos = new HashSet<Object>();
String obj1 = "JAVA";
int obj2 = 5;
Boolean obj3 = new Boolean(true);
objetos.add(obj3);
objetos.add(obj1);
objetos.add(obj2);
objetos.add(obj3);
for(Object object : objetos) {
System.out.print(object);
```

a. Error en tiempo de ejecución.
b. Se muestran por pantalla *JAVA 5* y *true* en un orden no determinado.
c. Se muestran por pantalla *JAVA 5* y *true* en el orden exacto en el que fueron insertadas en la colección.
d. Se muestran por pantalla *JAVA 5* y *true* en un orden no determinado y, además, "true" se muestra dos veces.

Pregunta: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
if(" String ".trim() == "String")
    System.out.println("Igual");
else
    System.out.println("No Igual");
```

a. El código compilará e imprimirá "Igual".
b. El código compilará e imprimirá "No Igual".
c. El código provocará un error de compilación.
d. El código provocará un error en tiempo de ejecución.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Un objeto es inmutable si su contenido o su estado no puede ser cambiado una vez que se ha creado.
b. Un objeto de tipo String puede ser modificado una vez que está creado, por tanto no es un ejemplo de objeto inmutable.
c. La clase String tiene un método de nombre trim que permite modificar caracteres en cualquier posición de una cadena.
d. Como regla general, las cadenas de texto de tipo String se suelen comparar mediante el operador "==".

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. La interfaz de una clase describe lo que hace la clase y cómo puede usarse pudiendo mostrar parte de su implementación.
b. Un mapa es una colección que almacena entradas de ternas de valores llave/valor/posición.
c. La documentación de una clase debe ser suficientemente detallada como para que otros programadores puedan usar la clase sin necesidad de leer su implementación.
d. Los modificadores de acceso definen las restricciones de uso de un objeto para determinados métodos, constructores o campos.

Pregunta: ¿Cuál es la salida que se obtiene al ejecutar este programa?

```
public class Test {
    public int aMethod() {
        static int i = 0;
        i++;
        return i;
    }
    public static void main(String args[]) {
        Test test = new Test();
        int i = test.aMethod();
        int j = test.aMethod();
```

```
            System.out.println(j);
        }
    }
```

a. 0
b. 1
c. 2
d. La compilación falla.

Pregunta: ¿Qué ocurrirá al compilar y ejecutar el siguiente código?

```
public class MiClase{
 static int variableEstatica;
 public static void main(String argv[]){
 System.out.println(variableEstatica);
 }
}
```

a. Error en tiempo de ejecución. La variable "variableEstatica" no ha sido inicializada
b. Se mostrará en pantalla null
c. Se mostrará en pantalla 1
d. Se mostrará en pantalla 0

Pregunta: De acuerdo a la bibliografía básica, el que una variable en una clase sea estática implica
a. Hace falta crear un objeto para usarla.
b. Cualquier objeto de esa clase puede modificar su valor.
c. Todos los objetos tienen una copia de la variable.
d. Que es una variable global y se puede usar en cualquier parte de la aplicación.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes afirmaciones es correcta:
a. Un objeto de tipo String puede ser modificado una vez que está creado, por tanto no es un ejemplo de objeto inmutable
b. La clase String tiene un método de nombre trim que permite modificar caracteres en cualquier posición de una cadena
c. Como regla general, las cadenas de texto de tipo String se suelen comparar mediante el operador ==
d. Un objeto es inmutable si su contenido o su estado no puede ser cambiado una vez que se ha creado

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:
a. En Javadoc la etiqueta @param indica el número de parámetros del método
b. En Javadoc la etiqueta @deprecated indica que el valor devuelto por el método puede contener errores
c. En Javadoc la etiqueta @see indica una referencia cruzada
d. En Javadoc la etiqueta @throws indica el modo en que debe ser lanzado un método

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes afirmaciones es correcta:

a. Un objeto es inmutable si su contenido o su estado no puede ser cambiado una vez que se ha creado
b. Un objeto de tipo String puede ser modificado una vez que está creado, por tanto no es un ejemplo de objeto inmutable
c. La clase String tiene un método de nombre trim que permite modificar caracteres en cualquier posición de una cadena
d. Como regla general, las cadenas de texto de tipo String se suelen comparar mediante el operador ==

Pregunta: ¿Cuál sería la salida del siguiente código?

```
int x = 0;
 String s = null;
 if(x == s) {
      System.out.println("Exito");
 } else {
      System.out.println("Fracaso");
 }
```

a. Exito
b. Fracaso
c. Error de compilación en la línea 4
d. Error de compilación en la línea 5

Pregunta: Dada la siguiente declaración:

```
Map < String, Double > map = new HashMap < String, Double > ();
```

¿Cuál de las siguientes opciones es correcta?

a. `map.add( " pi " , 3.14159);`
b. `map.add( " e " , 2.71828D);`
c. `map.add( " log(1) " , new Double(0.0));`
d. Ninguna de las anteriores

Pregunta: Si se ejecuta el siguiente código, ¿qué se imprime por pantalla?

Número de Línea	Código
4	`String s = new String ("Bicycle");`
5	`int iBegin = 1;`
6	`char iEnd = 3;`
7	`System.out.println(s.substring(iBegin,iEnd));`

a. Bic
b. ic
c. icy
d. error: no existe un método del tipo substring(int, char)

Pregunta: Dado el siguiente fragmento de código:

Número de Línea	Código
4	`String cadena1 = "Halo";`
5	`String cadena2 = "HALO";`
6	`cadena1.toUpperCase();`

¿Cuál sería el resultado de evaluar: `if (cadena1.equals(cadena2))`?

a. true
b. false
c. Error en la expresión
d. Ninguna de las anteriores es correcta

Pregunta: ¿Cuál es el resultado de ejecutar el siguiente fragmento de código?

```
if ("String".toString() == "String")
     System.out.println("Igual");
else
```

```
        System.out.println("No Igual");
```

a. El código compilará e imprime "Igual"
b. El código compilará e imprime "No Igual"
c. El código compilará pero producirá un error de ejecución
d. El código no compilará

Pregunta: Una variable de clase, definida como `static` ...

a. No puede ser accedida desde otra clase
b. Si se modifica, lo hace para todas las instancias de la clase
c. Es de valor constante
d. Solo puede ser accedida desde clases del mismo paquete

CAPÍTULO 6. OBJETOS CON BUEN COMPORTAMIENTO

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Un encapsulamiento apropiado en las clases reduce el acoplamiento
b. El término acoplamiento describe cuánto se ajusta una unidad de código a una tarea lógica o a una entidad
c. El acoplamiento describe la conectividad de los propios objetos de una clase
d. Un sistema débilmente acoplado se caracteriza por la imposibilidad de modificar una de sus clases sin tener que realizar cambios en ninguna otra

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. La depuración es la actividad cuyo objetivo es determinar si una pieza de código produce el comportamiento pretendido
b. La prueba viene a continuación de la depuración
c. La depuración es una actividad dedicada a determinar si un segmento de código contiene errores
d. La depuración es el intento de apuntar con precisión y corregir un error en el código

Pregunta: ¿Cuál es el resultado de la ejecución de las siguientes líneas de código?

```
28. Integer i = 5;
29. switch(i) {
30. case 1: System.out.print(1); break;
31. case 3: System.out.print(3);
32. case 5: System.out.print(5);
33. case 7: System.out.print(7); break;
34. default: System.out.print("default");
35. }
```

a. 5
b. 57
c. 57default
d. Error de compilación en la línea 29

Pregunta: ¿Cuál es el resultado del siguiente código?

```
4. final char a = 'A', d = 'D';
5. char nota = 'B';
6. switch(nota) {
7.    case a :
8.    case 'B' :
9.         System.out.print("enhorabuena");
```

```
10.   case 'C' :
11.         System.out.print("aprobado");
12.          break;
13.   case d :
14.   case 'F' :
15.         System.out.print("not good");
16. }
```

a. enhorabuena
b. enhorabuenaaprobado
c. Error de compilación en la línea 4
d. Error de compilación en la línea 7

Pregunta: ¿Cuál es la salida del siguiente código?

```
1. public class Incognita {
2. public static int metodoIncognita(String input) {
3.    int count = 0;
4.    int length = input.length();
5.    int i = 0;
6.
7.    String lowercase = input.toLowerCase();
8.    while(i < length) {
9.          switch(lowercase.charAt(i)) {
10.               case 'a':
11.               case 'e':
12.               case 'i':
13.               case 'o':
14.               case 'u':
15.                    count++;
16.         }
17.         i++;
18.   }
19. return count;
20. }
21.
22. public static void main(String [] args) {
23.   int x = metodoIncognita("Otorrinolaringologo");
24.   System.out.print(x);
25. }
26. }
```

a. 0
b. 9
c. 19
d. 20

Pregunta: Termina la frase. Si todos los campos no finales de una clase se declaran como privados y, además, la clase contiene métodos públicos para modificar o consultar dichos campos, esto es un ejemplo de:

a. Encapsulamiento alto
b. Acoplamiento bajo
c. Cohesión alta
d. Una relación "es un"

Pregunta: Dada la siguiente clase Television:

```
public class Television {
```

```
      public int canal;
      private boolean estaEncendida;
      private int volumen;
      public void cambiarCanal(int nuevoCanal) {
            canal = nuevoCanal;
      }
      public int consultarCanal() {
            return canal;
      }
      public void encender() {
            estaEncendida = true;
      }
      public void apagar() {
            estaEncendida = false;
      }
      public void subirVolumen() {
            volumen += 1;
      }
      public void bajarVolumen() {
            volumen -= 1;
      }
      }
```

¿Qué podemos afirmar?

a. La clase está altamente encapsulada.
b. La clase está altamente acoplada
c. La clase tiene un grado de cohesión alto
d. La clase tiene un grado de cohesión bajo

CAPÍTULO 7. DISEÑAR CLASES

Pregunta : Dado el siguiente fragmento de código, indique cuál de los siguientes resultados es el resultado de su ejecución:

```
public class test {
   public static void main(String args[]) {
      int i, j=1;
      i = (j>1)?2:1;
      switch(i) {
        case 0: System.out.print(0); break;
        case 1: System.out.print(1);
        case 2: System.out.print(2); break;
        case 3: System.out.print(3); break;
      }
   }
}
```

a. 01
b. 12
c. 13
d. 23

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. El encapsulamiento apropiado en las clases reduce su acoplamiento.
b. El acoplamiento describe el encapsulamiento de las clases.
c. El encapsulamiento apropiado en las clases reduce su cohesión.
d. La cohesión de una unidad de código refleja su acoplamiento.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Un encapsulamiento apropiado en las clases reduce el acoplamiento.
b. El término acoplamiento describe cuánto se ajusta una unidad de código a una tarea lógica o a una entidad.
c. El acoplamiento describe la conectividad de los propios objetos de una clase.
d. Un sistema débilmente acoplado se caracteriza por la imposibilidad de modificar una de sus clases sin tener que realizar cambios en ninguna otra.

Pregunta: En una estructura switch, ¿en qué lugar tiene que ser colocado el bloque de sentencias "default"?

a. Antes de las diferentes sentencias case.
b. Después de todas las sentencias case.
c. Después de las sentencias case pero antes de la sentencia finally.
d. Puede colocarse en el lugar que se quiera.

Pregunta: ¿Qué ocurre si se compila y ejecuta el siguiente código?

Número de Línea	Código
4	`public class Clase {`
5	`public static void main (String []arguments) {`
6	`met (arguments); }`
7	`public void met (String []arguments) {`
8	`System.out.println(arguments);`
9	`System.out.println(arguments[1]); }`

a. Da un error de compilación porque no se puede hacer referencia al método no-estático met.
b. Da un error de compilación porque el método main no puede ser estático.
c. Da un error de compilación porque el array arguments no puede pasarse como parámetro al método met.
d. Da un error de ejecución porque en el acceso al array arguments nos salimos del rango de dicho array.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. La depuración es la actividad cuyo objetivo es determinar si una pieza de código produce el comportamiento pretendido.
b. La prueba viene a continuación de la depuración.
c. La depuración es una actividad dedicada a determinar si un segmento de código contiene errores.
d. La depuración es el intento de apuntar con precisión y corregir un error en el código.

Pregunta: Dado el siguiente código, indique cuál de las siguientes afirmaciones es cierta:

```
public class MiClase {
      int x = 2;
      float y = 4.3f;
      public static void main (String [] args) {
      for (int z = 1; z < x; z++ )
            System.out.println("Valor de y="+y);
      }
}
```

a. Se produce un error en tiempo de ejecución.
b. El código no compila.
c. Se imprime en pantalla "Valor de y=4.3"
d. Se imprime en pantalla "Valor de y=4.3000"

Pregunta: Según el texto de la bibliografía básica de la asignatura, cuando un objeto permite realizar un conjunto de tareas muy relacionadas entre sí, podemos afirmar que:

a. El objeto presenta una alta cohesión.
b. El objeto está muy acoplado.
c. El objeto está poco encapsulado.
d. El objeto presenta una baja cohesión.

Pregunta: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
class Test
{
    public static void main (String args [])
    {
        int n, c = 1, serie = 5;
        System.out.print ("Cantidad de terminos: ");
        n = 7;
        while (c <= n)
        {
            System.out.print ("," + serie);
            serie += 5;
            c++;
        }
    }
}
```

a. Cantidad de términos: 5,10,15,20,25,30,
b. Cantidad de términos: ,5,10,15,20,25,30
c. Cantidad de términos: ,5,10,15,20,25,30,35
d. Cantidad de términos: ,5,10,15,20,25,30,35,40

Pregunta: ¿Cuál es la salida del siguiente programa?

```
class Test {
    public static void main(String [] args) {
        Test p = new Test();
        p.start();
    }

    void start() {
        boolean b1 = false;
        boolean b2 = fix(b1);
        System.out.println(b1 + " " + b2);
    }
    boolean fix(boolean b1) {
        b1 = true;
        return b1;
    }
}
```

a. true true
b. false true
c. true false
d. false false

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Un encapsulamiento apropiado en las clases reduce el acoplamiento.
b. El término acoplamiento describe cuánto se ajusta una unidad de código a una tarea lógica o a una entidad.
c. El acoplamiento describe la conectividad de los propios objetos de una clase.
d. Un sistema débilmente acoplado se caracteriza por la imposibilidad de modificar una de sus clases sin tener que realizar cambios en ninguna otra.

CAPÍTULO 8. MEJORAR LA ESTRUCTURA MEDIANTE HERENCIA

Pregunta: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
1.  class Uno{
2.      protected Uno yoMismo(){ return this;}
3.  }
4.  class Dos extends Uno{
5.      public Dos yoMismo(){
6.          return super.yoMismo();
7.      }
8.  }
```

a. No hay errores en el código. El resultado sería una referencia a un objeto del tipo Uno
b. No hay errores en el código. El resultado sería una referencia a un objeto del tipo Dos
c. Incompatibilidad de tipos línea 6.
d. El método yoMismo de la clase Uno no es visible en línea 6.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es **FALSA** en relación a los métodos polimórficos:

a. Una variable polimórfica es aquella que puede almacenar objetos de diversos tipos.
b. Las llamadas a métodos en Java no son polimórficas.
c. El mismo método puede invocar en diferentes momentos diferentes métodos dependiendo del tipo dinámico de la variable usada para hacer la invocación.
d. Cada objeto en Java tiene un método `toString` que puede usarse para devolver un `String` de su representación.

Pregunta: Dado el siguiente código …

Número de Línea	**Código**
4	`public class testJunio {`
5	`public void setVar (int a, int b, float c) {`
6	`}`
7	`// INSERTAR CÓDIGO AQUÍ`
8	`}`

Y los siguientes métodos:

```
1  private void setVar (int a, float c, int b) { }
2  protected void setVar (int a, int b, float c) { }
3  public int setVar (float a, int b, int c) {return b;}
4  public int setVar (int a, int b, float c) {return a;}
5  protected float setVar (int a, int b, float c) {return c;}
```

Indique qué métodos permiten una sobrecarga del método `setVar` de manera correcta:

a. 1 y 2
b. 1 y 3
c. 3 y 5
d. 3 y 4

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Una superclase es una clase que es implementada por otra.
b. Una subclase es una clase que implementa a otra clase.
c. Las clases que están vinculadas mediante una relación de herencia forman una jerarquía de herencia.
d. La herencia nos permite heredar pero no reutilizar en un nuevo contexto clases que fueron escritas previamente.

Pregunta: ¿Qué ocurrirá al compilar y ejecutar el siguiente código?

```
class Padre {}
class ClaseHija extends Padre {}
class ClaseHija2 extends Padre {}
public class Test{
    public static void main(String argv[]){
  Padre b=new Padre ();
  ClaseHija s=(ClaseHija) b;
  System.out.print("Ejecutando Aplicación");
   }
}
```

a. Compilará y se ejecutará sin problemas
b. Error de Compilación
c. Excepción en tiempo de ejecución
d. Excepción en tiempo de ejecución y luego mostrará el mensaje "Ejecutando Aplicación"

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Una superclase es una clase que es implementada por otra.
b. Una subclase es una clase que implementa a otra clase.
c. La herencia nos permite heredar pero no reutilizar en un nuevo contexto clases que fueron escritas previamente.
d. Las clases que están vinculadas mediante una relación de herencia forman una jerarquía de herencia.

Pregunta: Dadas las siguientes definiciones de clases:

```
class ClasePadre{}
class ClaseHija1 extends ClasePadre {}
class ClaseHija2 extends ClasePadre {}
```

y las siguientes instanciaciones:

```
ClasePadre var0 = new ClasePadre();
ClaseHija1 var1 = new ClaseHija1();
ClaseHija2 var2 = new ClaseHija2();
ClasePadre var3 = new ClaseHija1();
ClasePadre var4 = new ClaseHija2();
```

¿Cuál de las asignaciones es válida?

a. var0 = var1;
b. var2 = (ClaseHija2)var1;
c. var2 = var4;
d. var1 = var2;

Pregunta 15: Según el código siguiente ¿Qué se visualizará en pantalla?

```
class ClaseA{
   public ClaseA ( int x ){
      System.out.print("ClaseA-" + x);}
}

class ClaseB extends ClaseA{
   public ClaseB( ){
```

```
        super(6);
        System.out.print(" ClaseB-");}
}

public class ClasePrincipal{
   public static void main(String[] args) {
      ClaseB objB1=new ClaseB();
      ClaseB objB2;
      System.out.println(" FIN"); }
}
```

a. ClaseA-6 ClaseB- FIN
b. ClaseB- ClaseA-6 FIN
c. Hay un error en la clase B. La sentencia “super(6);” no puede ser la primera en el constructor
d. Hay en error en la clase ClasePrincipal. Falta el new en “ClaseB objB2;”

Pregunta: Dado el siguiente fragmento de código que pretende mostrar un ejemplo de sobrescritura:

```
class BaseClass {
        private float x = 1.0f ;
        protected float getVar () {return x;}
}

class Subclass extends BaseClass {
        private float x = 2.0f;
        //Insertar código aquí
}
```

Indique cual de las siguientes opciones completaría el código anterior para dar lugar a un ejemplo correcto de sobrescritura:

a. float getVar ( ) { return x;}
b. public float getVar ( ) { return x;}
c. float double getVar ( ) { return x;}
d. public float getVar (float f) { return f;}

Pregunta: Dado el siguiente código …

Número de Línea	Código
4	`public class testJunio {`
5	`public void setVar (int a, int b, float c) {`
6	`}`
7	`// INSERTAR CÓDIGO AQUÍ`
8	`}`

Y los siguientes métodos:

```
1   private void setVar (int a, float c, int b) { }
2   protected void setVar (int a, int b, float c) { }
3   public int setVar (float a, int b, int c) {return b;}
4   public int setVar (int a, int b, float c) {return a;}
5   protected float setVar (int a, int b, float c) {return c;}
```

Indique qué métodos permiten una sobrecarga del método `setVar` de manera correcta:

a. 1 y 2
b. 1 y 3
c. 3 y 5
d. 3 y 4

Pregunta: Dada la siguiente clase TV:

```
1. public class TV {
2. private String marca;
3. private String modelo;
4.
5. public TV(String marca, String modelo) {
6.     this.marca = marca;
7.     this.modelo = modelo;
8. }
9.
10. public boolean equals(TV other) {
11.    return marca.equals(other.marca) &&
12.    modelo.equals(other.modelo);
13. }
14.
15. }
```

¿Cuál sería el resultado de ejecutar el siguiente codigo?

```
TV a = new TV("Philips", "42PFL5603D");
TV b = new TV("Philips", "42PFL5603D");
if(a.equals(b)) {
System.out.println("iguales");
} else {
System.out.println("no son iguales");
}
```

a. iguales
b. no son iguales
c. Error de compilacion en la linea 11
d. Excepcion en tiempo de ejecucion en la linea 15

CAPÍTULO 9. ALGO MÁS SOBRE HERENCIA

Pregunta: Dado el siguiente fragmento de código, indique cuál es el resultado de su compilación:

```
1. class Parent {
2.    Double get() {
3.       return 1.0;
4.    }
5. }
6. class Child extends Parent {
7.    Integer get() {
8.       return 2;
9.    }
10.  }
```

a. Éxito.
b. get() en Child no puede extender get() en Parent, tipos del retorno son incompatibles.
c. get() en Child no puede extender get() en Parent, no son clases públicas.
d. get() en Child ya definido en Parent.

Pregunta: Sean "Mamifero" y "Gato" dos clases que mantienen una relación de herencia padre-hijo. ¿Qué habría que modificar en el siguiente código para que sea correcto y por qué?

```
Animal a;             /* Linea 1 */
Gato b;               /* Linea 2 */
a = new Animal ();    /* Linea 3 */
b = a;                /* Linea 4 */
```

a. Nada. Es correcto.

b. No se puede asignar un objeto a otro de otra clase, luego cambiamos la línea 2: Animal b; .
c. Es necesario explicitar el tipo cuando asignamos un objeto a otro objeto perteneciente a una clase hija, luego cambiamos la línea 4: b = (Gato) a; .
d. Es necesario explicitar el tipo y crear una nueva instancia cuando asignamos un objeto a otro objeto perteneciente a una clase hija, luego cambiamos la línea 4: b = new (Gato) a; .

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes opciones declarará un método en una clase que fuerza a una subclase a implementarlo:

a. `static void methoda (double d1) {}`
b. `public native double methoda();`
c. `abstract public void methoda();`
d. `protected void methoda (double d1){}`

Pregunta: Dado el siguiente código

```
String c1=new String(”Hola”);
String c2=new String(”Mundo”);
if (……)
      System.out.println(“Ambas cadenas son iguales”);
else
      System.out.println(“Ambas cadenas no son iguales”);
```

¿Cuál de las siguientes opciones debería ponerse en la línea de puntos para llevar a cabo la comparación de las cadenas c1 y c2 en función de la salida proporcionada por el programa?

a. c1==c2
b. c1.equals(c2)
c. c1.compareTo(c2)>=0
d. c1=c2

Pregunta: Dada la siguiente definición de clase

```
public class TV {
      private String marca;
      private String modelo;
      public TV(String marca, String modelo) {
            this.marca = marca;
            this.modelo = modelo;
      }
      public boolean equals(Object t) {
            TV television=(TV)t;
            return marca.equals(television.marca)&&
modelo.equals(television.modelo);
      }
       public int hashCode() {
             return marca.length() * 10 + modelo.length();
      }
}
```

¿Cuál sería el resultado visualizado al ejecutar el siguiente código en un método main?

```
            TV tv1 = new TV("Sony", "Bravia");
            TV tv2 = new TV("Sony", "aivarB");

            if(tv1.equals(tv2)) {
            System.out.println("los televisores son iguales");
            } else {
            System.out.println("los televisores no son iguales");
            }
```

a. Los televisores son iguales.

b. Los televisores no son iguales.
c. Error de compilación
d. Error en tiempo de ejecución

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es **FALSA** en relación a los métodos polimórficos:

a. Una variable polimórfica es aquella que puede almacenar objetos de diversos tipos.
b. Las llamadas a métodos en Java no son polimórficas.
c. El mismo método puede invocar en diferentes momentos diferentes métodos dependiendo del tipo dinámico de la variable usada para hacer la invocación.
d. Cada objeto en Java tiene un método `toString` que puede usarse para devolver un `String` de su representación.

CAPÍTULO 10. MÁS TÉCNICAS DE ABSTRACCIÓN

Pregunta: ¿Qué mecanismo usa Java para implementar herencia múltiple?

a. En Java no se permite la herencia múltiple de clases, ni tampoco la implementación múltiple de interfaces.
b. En Java no se permite la herencia múltiple de clases, pero sí la implementación múltiple de interfaces.
c. En Java se permite la herencia múltiple de clases, pero no la implementación múltiple de interfaces.
d. En Java se permite la herencia múltiple de clases, y también la implementación múltiple de interfaces.

Pregunta: ¿Qué interfaz proporciona la capacidad de almacenar objetos usando un valor llave?

a. Java.util.Map.
b. Java.util.Set.
c. Java.util.List.
d. Java.util.Collection.

Pregunta: Dados los siguientes fragmentos de código ¿Cuál de ellos asociaría a una Interfaz en Java?

a. public class Componente interface Product
b. Componente cp = new Componente (interfaz)
c. public class Componente implements Printable
d. Componente cp = new Componente.interfaz

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes afirmaciones es correcta:

a. Un mapa es una colección que almacena pares llave/valor como entradas.
b. Un mapa es una colección que almacena tríos llave/índice/valor como entradas.
c. Un mapa es una colección que almacena pares índice/valor como entradas.
d. Un mapa es una colección que almacena tríos índice/posición/valor como entradas.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes opciones declarará un método en una clase que fuerza a una subclase a implementarlo:

a. protected void metodoPI (double d1){}
b. abstract public void metodoPI ();
c. static void metodoPI (double d1) {}
d. public native double metodoPI ();

Pregunta: Dado el siguiente fragmento de código que pretende mostrar un ejemplo de sobrescritura, indique cuál de las siguientes opciones completaría el código para dar lugar a un ejemplo correcto de sobrescritura:

Número de Línea	Código
4	`class BaseClass {`
5	`private float x = 1.0f ;`
6	`protected float getVar () {return x;}`
7	`}`

```
8       class Subclass extends BaseClass {
9            private float x = 2.0f;
10           //Insertar código aquí
11      }
```

a. `float getVar ( ) { return x;}`
b. `public float getVar ( ) { return x;}`
c. `float double getVar ( ) { return x;}`
d. `public float getVar (float f ) { return f;}`

Pregunta: Dado el siguiente código:

```
public class ClaseUno {
     ClaseUno obj;
     ClaseUno() { }
     ClaseUno(ClaseUno m) { obj = m; }

     void inicializar() { System.out.print("Inicializando. ");}
}

public class Test {
public static void main(String[] args) {
     ClaseUno obj1 = new ClaseUno();
     ClaseUno obj2 = new ClaseUno(obj1);
     obj2.inicializar();
     ClaseUno obj3 = obj2.obj;
     obj3.inicializar();
     ClaseUno obj4 = obj1.obj;
     obj4.inicializar();
     }
}
```

Podemos afirmar que:

a. Se mostrará el mensaje "Inicializando. Inicializando. Inicializando."
b. Se mostrará el mensaje "Inicializando. Inicializando".
c. Se mostrará el mensaje "Inicializando. Inicializando" seguido de una excepción.
d. Se mostrará el mensaje "Inicializando. Inicializando. Inicializando." seguido de una excepción

Pregunta: Indique cuál de las siguientes afirmaciones es cierta:

a. Una interfaz puede implementar alguno de los métodos que declara.
b. Una interfaz puede declarar variables de instancia o de clase.
c. Cuando una clase implementa una interfaz específica no hace falta que implemente todos los métodos que ésta declara.
d. Una clase puede implementar más de una interfaz al mismo tiempo.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes opciones declarará un método en una clase que fuerza a una subclase a implementarlo:

a. static void methoda (double d1) {}
b. public native double methoda();
e. abstract public void methoda();
c. protected void methoda (double d1){}

Pregunta: Dado el siguiente código:

```
30. Set < Object > objetos = new HashSet < Object > ();
31. String one = "hola";
32. int two = 2;
33. Boolean three = new Boolean(true);
34. objetos.add(one);
35. objetos.add(two);
36. objetos.add(three);
```

```
37. objetos.add(three);
38. for(Object objeto : objetos) {
39.    System.out.print(objeto);}
```

¿Cuál de las siguientes afirmaciones es cierta?

a. La salida es hola, 2 y true en un orden no determinado.
a. La salida es hola, 2, true y true en un orden no determinado.
b. Error de compilación en la línea 35.
c. Excepción en tiempo de ejecución en la línea 37.

Pregunta: Dadas las siguientes definiciones de clase y de interfaz:

```
1. //Legible.java
2. public interface Legible {
3.     public void leer();
4.     public int MAX_LENGTH = 10;
5. }
```

```
1. //MiLector.java
2. public class MiLector implements Legible {
3.     public void leer() {
4.            Legible.MAX_LENGTH = 25;
5.            System.out.println(Legible.MAX_LENGTH);
6.     }
7. }
```

¿Cual sería el resultado de ejecutar la siguiente línea de código?

```
new MiLector().leer();
```

a. 25
b. 10
c. Error de compilación en la línea 4 de Legible.java
d. Error de compilación en la línea 4 de MiLector.java

Pregunta: ¿Cual es el resultado del siguiente codigo?

```
1. public abstract class A {
2. private void doSomething() {
3.     System.out.println("A");
4. }
5.
6. public static void main(String [] args) {
7.     A a = new B();
8.     a.doSomething();
9. }
10. }
11.
12. class B extends A {
13. protected void doSomething() {
14. System.out.println("B");
15. }
16. }
```

a. A
b. B
c. Error de compilacion en la linea 7
d. Error de compilacion en la línea 8

Pregunta: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
public class TestSet {
  static void add(Set set) {
```

```
        set.add("Hola");
        set.add(1);
        System.out.println(set.size());
    }
    public static void main(String[] args) {
        Set<String> set = new HashSet<String>();
        add(set);
    }
}
```

a. 0
b. 1
c. 2
d. NullPointerException

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

a. Si una clase tiene algún método abstracto hay que declararla como abstracta.
b. Todos los métodos en una clase abstracta tienen que ser declarados como abstractos.
c. Una clase que hereda de una clase abstracta no tiene que implementar todos los métodos abstractos para no ser abstracta.
d. Una clase abstracta no puede implementar ninguna interface.

Pregunta: ¿Java permite herencia múltiple?

a. Sí, es una característica del lenguaje.
b. No, por definición no admite herencia múltiple y no puede implementarse de modo alguno.
c. No, pero puede implementarse mediante combinación de “extends” y de “implements interface”.
d. No, pero puede implementarse mediante combinación de “implements” y de “extends interface”.

Pregunta: Dado el siguiente fragmento, ¿podemos instanciar un objeto de esta clase?

Número de Línea	**Código**
4	`public abstract class ClaseAbstracta {`
5	`abstract void MetodoAbstracto(int a);`
6	`}`

a. Sí, creando una nueva clase que extienda a ClaseAbstracta.
b. Sí, creando una nueva clase que extienda a ClaseAbstracta e implemente el método MetodoAbstracto.
c. Si, las clases abstractas se pueden instanciar como cualquier otra clase si necesidad de extenderlas si se redefina el método MetodoAbstracto.
d. Ninguna respuesta anterior es correcta.

Pregunta: Necesita crear una clase que almacene como elemento base de la misma objetos únicos. No se necesita que guarden orden alguno, pero sí que no se repitan. ¿Qué interfaz sería la más apropiada para este fin?

a. Set.
a. List.
b. Map.
c. Vector.

Pregunta: En la definición de una interface en Java :

a. Es necesaria emplear la palabra clave abstract.
b. La signatura de los métodos de una interfaz tienen visibilidad public o private, pero no protected.
c. No se permiten campos constantes.

d. Aunque no se indique usando la palabra clave final, todos los campos son tratados como si así fuesen.

Pregunta: Un conjunto es una estructura:

a. Que almacena cada elemento individual una sola vez como mínimo. No mantiene un orden específico.
b. Que almacena cada elemento individual una sola vez como mínimo. Mantiene un orden específico.
c. Que almacena cada elemento individual una sola vez como máximo. No mantiene un orden específico.
d. Que almacena cada elemento individual una sola vez como máximo. Mantiene un orden específico.

CAPÍTULO 11. CONSTRUIR INTERFACES GRÁFICAS DE USUARIO

Pregunta: ¿Qué se mostrará en pantalla al ejecutar el siguiente código?

```
import java.awt.*;
import javax.swing.JFrame;

public class AppBoton extends JFrame{

  public static void main(String argv[]){
        AppBoton MiAppBoton=new AppBoton ();
 }

  public AppBoton(){
        Button boton1=new Button("BOTON 1");
        Button boton2=new Button("BOTON 2");
        add(boton1);
        add(boton2);
        setSize(100,100);
        setVisible(true);
 }
}
```

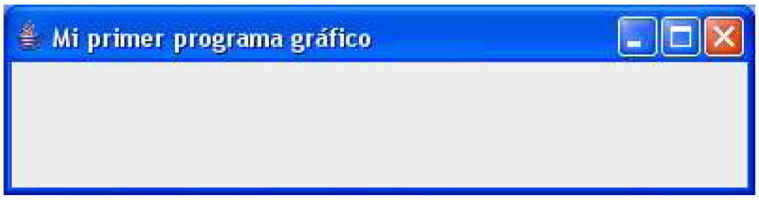
a. Dos botones, uno junto a otro ocupando todo el frame. En el botón de la izquierda aparecerá BOTON 1 y en el de la derecha aparecerá BOTON 2.
b. Un botón ocupando todo el frame con la etiqueta BOTON 1.
c. Un botón ocupando todo el frame con la etiqueta BOTON 2.
d. Dos botones en la parte superior del frame, uno de ellos con la etiqueta BOTON 1 y otro de ellos con la etiqueta BOTON 2.

Pregunta: La ejecución del siguiente fragmento de código …

Número de Línea	Código
4	`import javax.swing.*;`
5	`class PrimerFrame extends JFrame`
6	`{`
7	`public PrimerFrame()`
8	`{`
9	`setTitle("Mi primer programa gráfico");`
10	`setSize(400,100);`
11	`}`
12	`}`
13	`public class FrameTest`
14	`{`
15	`public static void main (String[] args)`
16	`{`
17	`JFrame frame = new PrimerFrame();`
18	`frame.setVisible (true);`
19	`}`

```
20          }
```

Da lugar al siguiente programa:



Pero este último programa tiene el problema de que cuando se cierra la ventana, a pesar de que dejamos de verla, el programa no finaliza su ejecución. De esta forma, para que el programa funcione correctamente, hemos de interceptar el evento que se produce cuando cerramos la ventana y hacer que el programa termine su ejecución en ese momento. Indique qué clase hemos de definir en este caso y asociárselo al `JFrame` del ejemplo:

a. `ActionListener`
b. `ComponentListener`
c. `WindowListener`
d. `ItemListener`

Pregunta: Si quisiera organizar los componentes de una interfaz gráfica de acuerdo a una tabla utilizaría como gestor de contenido:

a. GridLayout
b. BoxLayout
c. FlowLayout
d. BorderLayout

Pregunta: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
import java.awt.*;

public class TestFrame extends Frame
{
     public TestFrame()
     {
          setLayout(new GridLayout());
          for(int i = 1 ; i <= 4 ;++i)
          {
               add(new Button(Integer.toString(i)));
          }

          pack();
          setVisible(true);
      }

     public static void main(String args[])
     {
          TestFrame tf = new TestFrame();
     }
}
```

a. El código compila, su ejecución provoca que todos los botones aparezcan en una sola columna.
b. El código compila, su ejecución provoca que todos los botones aparezcan en una sola fila.
c. El código compila, su ejecución provoca que todos los botones se monten uno encima del otro y tan solo sea visible el último.
d. El código compila, pero se produce un error en tiempo de ejecución cuando se añaden los componentes.

Pregunta: Dada la siguiente definición de clase:

```
1. import java.awt.*;
2. import java.awt.event.*;
3.
4. public class MyWindow {
```

```
5.     private Frame frame = new Frame();
6.
7.     public void registerEvents() {
8.           WindowAdapter wa = new WindowAdapter() {
9.                 public void windowClosing(WindowEvent e) {
10.                      frame.setVisible(false);
11.                      frame.dispose();
12.                }
13.          };
14.          frame.addWindowListener(wa);
15.    }
16.}
```

¿Cual de las siguientes afirmaciones es cierta?

a. Hay un error de compilación en las líneas 10 y 11.
b. El objeto que se instancia en la línea 8 no tiene acceso al campo frame de la línea 5 porque este es privado.
c. El método de la línea 9 no se ejecuta nunca ya que deja de ser accesible a partir de la línea 15.
d. La clase anónima anidada de la línea 8 extiende la clase WindowAdapter .

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es **incorrecta** sobre las bibliotecas para la construcción de interfaces gráficas de usuario en Java:

a. AWT utiliza clases de Swing.
b. Swing utiliza clases de AWT.
c. Hay clases equivalentes en AWT y Swing.
d. Se identifican las clases de Swing con la letra J como prefijo.

Pregunta: ¿Qué se mostrará por pantalla cuando se intenta compilar y ejecutar el siguiente código?

```
import java.awt.*;
public class Butt extends Frame{
      public static void main(String argv[]){
            Butt MyBut=new Butt();
      }

      Butt(){
            Button HelloBut=new Button("Hola");
            Button ByeBut=new Button("Adios");
            add(HelloBut);
            add(ByeBut);
            setSize(300,300);
            setVisible(true);
      }
}
```

a. Dos botones juntos ocupando todo el frame. Hola en la izquierda y Adios en la derecha.
b. Un botón ocupando todo el frame diciendo Hola.
c. Un botón ocupando todo el frame diciendo Adios.
d. Dos botones en la parte superior del frame diciendo uno Hola y el otro Adios.

Pregunta: En lo que se refiere a las clases internas anónimas, se puede afirmar que:

a. Suelen emplearse en los lugares en los que se requiere la implementación de una sola instancia.
b. Siempre se hará referencia la instancia mediante su supertipo.
c. Permiten definir una clase y crear una instancia de ella, todo en un solo paso.
d. Todas las anteriores son correctas.

Pregunta: Una clase interna:

a. Puede acceder a los campos y métodos públicos y protegidos de la clase envolvente, pero no privados.
b. Puede acceder a los campos y métodos públicos de la clase envolvente, pero no a privados ni a protegidos.
c. Puede acceder a los campos y métodos públicos y privados de la clase envolvente.
d. No puede acceder a los campos y métodos privados de la clase envolvente.

Pregunta: Cuando queremos que un objeto oiga eventos de acción disparados por el usuario, el objeto tiene que implementar la interfaz ...

a. ActionEvent.
b. ActionListener.
c. ListenerAction.
d. ListenerEvent.

CAPÍTULO 12. MANEJO DE ERRORES

Pregunta: Respecto a las excepciones en Java, podemos afirmar …

a. Todas las subclases de la clase estándar de Java `RunTimeException` son excepciones comprobadas.
b. Todas las subclases de la clase estándar de Java `Exception` son excepciones comprobadas.
c. `Error` es una subclase directa de `Throwable`, mientras que `Exception` es una subclase directa de `Error`.
d. Tanto `Error` como `Exception` son subclases directas de `Throwable`.

Pregunta: ¿Qué código hay que añadir en la posición indicada en el código para que compile?

```
public class ExceptionTest {
    class TestException extends Exception {}
    public void runTest() throws TestException {}
    public void test() /* Codigo a añadir */  {
        runTest();
    }
}
```

a. No hay que añadir código alguno.
b. throws Exception
c. catch (Exception e)
d. throws RuntimeException

Pregunta: En el siguiente fragmento de código hemos definido la ejecución de cinco bloques. Estos bloques se ejecutarán dependiendo de las excepciones que se produzcan en cada caso.

```
// Bloque1
try{
      // Bloque2
}catch (ArithmeticException e) {
      // Bloque3
}finally{
      // Bloque4
}
// Bloque5
```

Indique cual de las siguientes afirmaciones es correcta:

a. El Bloque4 no se ejecutará si se produce una excepción de tipo aritmético en el Bloque2
b. El Bloque4 se ejecutará antes de que la excepción producida por un acceso a un objeto *null* en el Bloque2 se propague hacia arriba
c. El Bloque4 no se ejecutará si se produce un acceso a un objeto *null* en el Bloque2
d. El Bloque4 se ejecutará antes que el Bloque3 si se produce una excepción de tipo aritmético en el Bloque2

Pregunta: En el siguiente fragmento de código hemos definido la ejecución de cinco bloques. Estos bloques se ejecutarán dependiendo de las excepciones que se produzcan en cada caso. Indique cuál de las siguientes afirmaciones es correcta:

Número de Línea	Código
4	`// Bloque1`
5	`try{`
6	`    // Bloque2`
7	`}catch (ArithmeticException e) {`
8	`    // Bloque3`
9	`}finally{`
10	`    // Bloque4`
11	`}`
12	`// Bloque5`

a. El Bloque4 no se ejecutará si se produce una excepción de tipo aritmético en el Bloque2
b. El Bloque4 no se ejecutará si se produce un acceso a un objeto nulo (null) en el Bloque2
c. El Bloque4 se ejecutará antes que el Bloque3 si se produce una excepción de tipo aritmético en el Bloque2
d. El Bloque4 se ejecutará antes de que la excepción producida por un acceso a un objeto nulo (null) en el Bloque2 se propague hacia arriba

Pregunta: ¿Cual es el resultado del siguiente programa?

```
1. public class ComparadorRaro {
2. private Integer x;
3.
4. public boolean compare(int y) {
5.     return x == y;
6. }
7.
8. public static void main(String [] args) {
9.     ComparadorRaro u = new ComparadorRaro();
10.    if(u.compare(21)) {
11.           System.out.println(”true”);
12.    } else {
13.           System.out.println(”false”);
14.    }
15. }
16. }
```

a. true
b. false
c. Error de compilacion en la línea 5.
d. La línea 5 lanza una excepción NullPointerException

Pregunta: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
public class test {
    public static void main(String args[]) {
        int i=1, j=1;
        try {
            i++;
            j--;
            if(i == j)
                i++;
        }
        catch(ArithmeticException e) {
            System.out.print(0);
        }
        catch(ArrayIndexOutOfBoundsException e) {
            System.out.print(1);
        }
```

```
        catch(Exception e) {
            System.out.print(2);
        }
        finally {
            System.out.print(3);
        }
        System.out.print(“,4”);
    }
}
```

a. 0,4
b. 1,4
c. 2,4
d. 3,4

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes definiciones de un método m, que lanza IOException, y que devuelve void, es correcta:

a. void m() throws IOException {}
b. void m() throw IOException {}
c. void m(void) throws IOException {}
d. void m() {} throws IOException

Pregunta: Dado el siguiente fragmento de código, indique cuál es la salida de su compilación/ejecución:

```
1. String nombre = null;
2. File file = new File("/folder", nombre);
3. System.out.print(file.exists());
```

a. true
b. false
c. NullPointerException en línea 2.
d. NullPointerException en línea 3.

Pregunta: ¿Cuál de las siguientes es una característica de la clase java.lang.Exception?

a. private.
b. extends Throwable.
c. implements Throwable.
d. final.

CAPÍTULO 13. DISEÑAR APLICACIONES

Pregunta: Según el texto de la bibliografía básica de la asignatura, un prototipo es …

e. Una versión de la aplicación en la que se simula una parte de ella, en vías a experimentar con las restantes partes.
f. Una versión de la aplicación en la que se simulan varias partes, en vías a experimentar con una de sus partes.
g. Una versión de la aplicación en la que se simulan varias partes, en vías a experimentar con las restantes partes.
h. Ninguna de las anteriores

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta en relación a la programación por parejas:

a. Consiste en programar una clase por duplicado con el objetivo de depurar los errores más fácilmente.
b. Es una manera de producir código, opuesta a la programación extrema en la que un solo programador desarrolla las clases asignadas.
c. Era una técnica de programación tradicional que las empresas eliminaron para reducir costes.
d. Es uno de los elementos de una técnica que se conoce como programación extrema.

Pregunta: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta en relación a la programación por parejas:

a. Consiste en programar una clase por duplicado con el objetivo de depurar los errores más fácilmente.
b. Es una manera de producir código, opuesta a la programación extrema en la que un solo programador desarrolla las clases asignadas.
c. Era una técnica de programación tradicional que las empresas eliminaron para reducir costes.
d. Es uno de los elementos de una técnica que se conoce como programación extrema.

Pregunta: ¿Qué incluye como mínimo la descripción de un patrón?

a. Un nombre, una descripción del problema, una descripción de la solución y las consecuencias del uso del patrón.
b. Un nombre, una clase, una descripción del problema y las consecuencias del uso del patrón.
c. Una clase, una descripción del problema, una descripción de la solución y las consecuencias del uso del patrón.
d. Un nombre, una clase, una descripción de la solución y las consecuencias del uso del patrón.