

1.- Tenemos la clase coche:

```
class Coche {  
  
    int velocidad;  
  
    public Coche() {  
        velocidad = 0;  
    }  
}
```

Añade los siguientes métodos:

- `int getVelocidad()` // Para devolver la velocidad actual
- `void acelera(int mas)` // Para aumentar la velocidad
- `void frena(int menos)` // Para disminuir la velocidad

2.- Realiza una clase Convertidor que convierte entre euros y dólares y viceversa. Codifica los métodos `eurosToDolares` y `dolaresToEuros`. La clase debe tener:

- Un constructor por defecto, el cual establecerá el cambio euro-dolar en 1.36.
- Un constructor el cual permitirá configurar el cambio euro-dolar.

3.- Verdadero o Falso:

1. Las clases que forman parte de un paquete deben derivar todas ellas de una misma superclase.
2. En una clase Java los atributos pueden ser tipos primitivos o bien pueden ser objetos de otra clase.
3. En un programa OO, se definen las clases dotándolas de estado y comportamiento y cuando se ejecute el programa, se crearán los objetos.
4. La encapsulación significa que al proporcionar los ficheros `.class`, el programador no tiene por qué proporcionar los ficheros `.java` donde reside todo el código.
5. Un paquete es un conjunto de clases relacionadas entre sí.
6. La abstracción en POO es una de las propiedades fundamentales de la misma mediante la cual los objetos se ven según su comportamiento externo.
7. El polimorfismo permite crear varios métodos con el mismo nombre y el mismo comportamiento.
8. Las clases se escriben en ficheros de texto. El nombre del fichero puede ser cualquiera, únicamente tiene que tener extensión `.java`.

4.- La clase Festivo representa un día festivo del año. Esta clase tiene 3 propiedades: nombre, mes y día.

- a) Escribe un constructor que permita inicializar las 3 propiedades de la clase.
- b) Escribe un método `esMismoMes` que compare 2 instancias de la clase y devuelva verdadero o falso.
- c) Escribe un método "media" que dado un array de Festivo calcule la media de todas las propiedades "día".

5.- Escribir una clase Cajero que me permita realizar las operaciones más comunes de un cajero automático.

6.- Escribir una clase Pez que contenga una propiedad "numeroPeces" de tipo entero que sirva para llevar la cuenta del número de peces que se han creado. Añadir un método imprimirContador() que imprima el número de instancias. Crear varias instancias de la clase y llamar al método imprimirContador() en cada una de ellas. Observar el resultado.

7.- Qué mostrará la salida de este programa?

```
public class Bebe {  
  
    public Bebe (int i) {  
        this("Soy un bebe consentido");  
        System.out.println("Hola, tengo " + i + " meses");  
    }  
  
    public Bebe (String s) {  
        System.out.println( s );  
    }  
  
    void berrea() {  
        System.out.println("Buaaaaaaaaaaaa");  
    }  
  
    public static void main(String[] args) {  
        new bebe (8).berrea();  
    }  
}
```

8.- Un centro cultural se dedica al préstamo de dos tipos de materiales de préstamo: discos y libros. Para los dos se guarda información general, como su código identificativo, el título y el autor. En el caso de los libros, almacenamos también su número de páginas y un capítulo de muestra, y para los discos el nombre de la discográfica.

También podemos encontrar una serie de documentos con las normas e información sobre el centro de los que guardamos su título, fecha de publicación y texto. Tanto estos documentos como los libros se podrán imprimir (en el caso de los libros se imprimirá el título, los autores, y el capítulo de muestra, mientras que de los documentos se imprimirá su título, su fecha de publicación y su texto). Crear un proyecto que contenga las clases para representar esta información.

9.- En la empresa manejamos pedidos de cliente y pedidos internos.

Todos los pedidos tienen un número de pedido. Los pedidos de cliente tienen además un núm. de cliente y un valor "enviado" que nos indica si ya ha sido enviado a los clientes o no.

Los pedidos de cliente se envían al almacén, que es el que se encarga de enviar los pedidos.

También manejamos facturas: facturas de clientes y facturas internas (entre centros de la misma empresa), las facturas de cliente se envían a la contabilidad para ser contabilizadas.

Cualquier factura o pedido puede ser enviado a la cola de impresión para ser impresas. Implementar las clases e interfaces necesarias para representar este modelo.

10.- Crear una clase Vehiculo con las siguientes propiedades y métodos:

```
int velocidad;  
double precioTarifa;  
String color;  
double getPrecioVenta();
```

- Crear una subclase llamada Camión con:

```
int peso;  
double getPrecioVenta() {  
    // Si el peso es mayor que 2000 aplicar un 10% sobre precioTarifa, en otro caso  
    aplicar el 20%  
}
```

- Crear una subclase de Vehiculo llamada Coche:

```
int anho;  
int descuentoFabricante;  
double getPrecioVenta() {  
    // Sobre el precio de venta “calculado en la clase Vehiculo” descontar el  
    descuentoFabricante.  
}
```

- Crear una subclase de Coche llamada Sedan:

```
float longitud;  
double getPrecioVenta() {  
    // Si la longitud es mayor que 5 hacer un descuento del 10%, en otro caso del 5%  
}
```

- Crear un constructor para inicializar las respectivas propiedades de cada clase.

- Crear una aplicación que cree una instancia de cada clase y muestre el precio de venta de cada instancia.

11.- Crear una clase Fecha que tenga 3 propiedades día, mes, año y un método:

```
int compareTo(Fecha otraFecha) {  
    /* Devuelve:  
        -1 si es anterior que otraFecha  
        0 si son iguales  
        1 si es posterior que otraFecha  
    */  
}
```

Crear una clase Persona con una propiedad nombre y una fecha de nacimiento de tipo Fecha.

```
Realizar un método int compareTo(Persona otraPersona) {  
    /* Devuelve:  
        -1 si es menor que otraPersona  
        0 si son iguales  
        1 si es mayor que otraFecha  
    */  
}
```

12.- Crear una clase Equipo con un constructor que permita asignarle su nombre y un método getNombre que devuelva su nombre.

Crear una clase Jugador con las propiedades nombre y numeroGoles. Implementar los métodos: getNombre, getGoles.

A la clase Equipo añadir:

- Un método anadirJugador que permita anadir un jugador al equipo y un método imprimirJugadores() que imprima los nombres de los jugadores del Equipo.
- El método goles() que devuelva el número total de goles marcados por sus integrantes.
- Un método que devuelva el Jugador que más goles haya marcado.

13.- Añadir los métodos equals(), hashCode() y toString() a las clases del ejercicio 11.

14.- Añadir los métodos equals(), hashCode() y toString() a las clases del ejercicio 12.

15.- Cual es la diferencia entre POO y programación estructurada (o procedural)?

A – La POO es más rápida que la estructurada.

B – La programación estructurada trata un programa como una secuencia de acciones, mientras que un programa realizado según POO se descompone en un grupo de entidades llamadas objetos que tienen un estado y un comportamiento.

C- POO es más potente y capaz de resolver problemas más complejos que la programación estructurada.

16.- What is an object? How is an object different from a class?

A - Objects are used in object-oriented programming and classes are used in class-oriented programming.

B - A class is an instance of an object. One object can be used to create many classes.

C - An object is not encapsulated and a class is encapsulated, making classes more powerful and reusable than objects.

D - An object is a kind of class that does not contain any behavior (methods).

E - An object is an entity that encapsulates related data and behavior, while a class is the blueprint for a type of objects.

17.- What is the state and behavior of a String object?

1. The state is the location the text came from, such as a file or from the Scanner, and the behavior is when you save that string to a new location, such as to a new file.
2. The state is the string itself, and the behavior is what you do with it, such as putting it in an array or list.
3. The state is the memory occupied by the object, and the behavior is what you do with that memory, such as storing characters in it.
4. The state is its sequence of characters, and the behavior is its methods, such as length and indexOf.
5. The state is methods that return values, such as length or charAt, and the behavior is methods that change the string, such as toUpperCase or substring.

18.-

Which of the following are differences between a field and a parameter?

1. A field's scope is throughout the class, while a parameter's scope is limited to the method.
2. Field syntax differs because they can be declared with the 'private' keyword.
3. Parameters must be primitive types of values, while fields can be objects.
4. You can only have one field per class, while you can have as many parameters as you want.
5. Fields can store many values while parameters can store only a single value.
6. A field is a variable that exists inside of an object, while a parameter is a variable inside a method whose value is passed in from outside.
7. Fields are constant and can be set only once, while parameters change on each call.
8. A field takes up more memory in the computer than a parameter does.

19.- El siguiente programa imprime 4 líneas por pantalla. Cuáles?

```
public class Mystery {

    public static void main(String[] args) {
        int a = 7;
        int b = 9;
        Point p1 = new Point(2, 2);    // La declaración de este constructor es
        Point p2 = new Point(2, 2);    // public Poin(int x, int y) {...}
        addToXTwice(a, p1);
        System.out.println(a + " " + b + " " + p1.x + " " + p2.x);
        addToXTwice(b, p2);
        System.out.println(a + " " + b + " " + p1.x + " " + p2.x);
    }

    public static void addToXTwice(int a, Point p1) {
        a = a + a;
        p1.x = a;
        System.out.println(a + " " + p1.x);
    }
}
```

20.-

What is the meaning of the keyword 'this', and how can the keyword be used? Check all that apply.

1. It refers to the object on which a method or constructor has been called (sometimes called the "implicit parameter").
2. It is used to call one constructor from another.
3. It can be used to access or set an object's field values
4. It is used when one object wants to access data from a second object.
5. It is used in conjunction with the 'that' keyword when the programmer wants to write a parallel "this and that" algorithm.
6. It is required when a class has more than one constructor.
7. It can be used to call the object's methods.

21.-

What is abstraction?

1. When alcohol was illegal in the United States in the early 20th century.
2. The ability to focus on a problem at a high level without worrying about the minor details.
3. When things are confusing and vague rather than simple and understandable.
4. The opposite of addition, as seen in mathematics.
5. A Java keyword that optimizes programs that do not use classes or objects.

(order shuffled)

How do objects provide abstraction?

1. By giving us more powerful pieces of data that have sophisticated behavior without having to manage and manipulate the data directly.
2. When one object converts into another type of object, it becomes abstract.
3. Public fields provide abstraction because they let the client directly view an object's data.
4. Objects provide abstraction through the .provideAbstraction() method.
5. Objects provide abstraction when they declare a lot of useful data inside them.

22.-

Which of the following is the correct syntax to indicate that class A is a subclass of B?

1. `public class A : super B {`
2. `public A(super B) {`
3. `public class B extends A {`
4. `public class A extends B {`
5. `public A implements B {`

