

C1. Los objetos se crean a partir de **clases**. La **clase** describe el tipo de objeto. Los objetos representan instancias individuales de la **clase**.

C2. Los objetos se comunican unos con otros a través de **mensajes**. Un objeto hace algo si invocamos a un **método** de ese objeto.

C3. La cabecera de un método se denomina **signatura**.

C4. Los **atributos** de un objeto definen su estado y los **métodos** definen su comportamiento.

C5. Los métodos pueden tener **parámetros** que proporcionan información adicional para que el método realice su tarea.

C6. Los parámetros tienen un **tipo**. El **tipo** indica la clase de valores que el parámetro puede tomar.

C7. Los métodos pueden devolver información vía un **valor de retorno (return)**.

C8. Objeto es sinónimo de **instancia**.

C9. ¿Qué es un programa orientado a objetos? ¿Qué son los mensajes?

- Un programa orientado a objetos es un conjunto de objetos que interactúan entre sí a través de mensajes. Los mensajes son los datos con los que trabajan los objetos.

C10. Diferencia entre clase y objeto. Pon un ejemplo.

- La clase define el tipo de objetos que tiene, es decir, una clase es como una plantilla de la cual se crearán objetos con un estado independiente.
- La clase Persona creará objetos Persona.

C11. Tengo un ordenador de marca XXXXX con 512Mb de RAM y 160 Gb de disco duro. Mi ordenador es capaz de ejecutar programas, informarme acerca de la cantidad de memoria que posee y acerca de la capacidad del disco duro. De qué estoy hablando, ¿de un objeto o de una clase? Identifica los atributos y su tipo y los métodos (servicios).

De un objeto, ya que tiene un estado.

- **Atributos:** marca, tamanoMemoria, tamanoDisco.
- **Métodos:** getMarca(), getTamanoMemoria(), getTamanoDisco(), ejecutarPrograma().

C12. El ordenador anterior ofrece también la posibilidad de aumentar la memoria en una determinada cantidad. Escribe la signatura del método anadirMemoria() indicando en caso de que los haya, parámetros y valor de retorno.

```
private int memoria;  
public void anadirMemoria(int memoria) {  
    this.memoria += memoria;  
}
```

C13. Pon un ejemplo de un atributo de tipo primitivo y otro de tipo referencia.

```
private int contador;  
private String alias;
```

C14. Imaginemos una clase Urna que modela una urna capaz de almacenar bolas de diferentes colores. Las bolas son objetos de la clase Bola. Sobre una urna se pueden añadir y extraer bolas. Define la signatura de los métodos anadirBola() y extraerBola() que añaden una nueva bola a la urna y devuelven la bola extraída de la urna respectivamente.

```
public class Urna  
{  
    public Urna() {  
    }  
    public void anadirBola(Bola bola) {  
    }  
    public Bola extraerBola() {  
        return bola;  
    }  
}
```

C15. Para los siguientes ejemplos de clases indica cuáles podrían ser sus atributos y el tipo de éstos:

- una clase TelefonoMovil cuyas instancias guardan el nº de teléfono móvil y la cantidad de dinero que todavía queda de la recarga

```
private String numeroTelefono;  
private double saldo;
```

- una clase Punto que modela a los puntos del espacio bidimensional

```
private double horizontal;  
private double vertical;
```

- una clase Hora cuyos objetos guardan la hora actual

```
private int segundo;  
private int minuto;  
private int hora;
```

- una clase Bombilla cuyas instancias guardan la potencia y el estado (si está encendida o apagada)

```
private int potencia;  
private boolean encendida;
```

- la clase Producto que modela el comportamiento de los productos de un almacén registrando para cada uno su identificador (una serie de caracteres) y la cantidad en stock

```
private String id;  
private int cantidad;
```

C16. Escribe ahora las firmas de alguno de los métodos que describen el comportamiento de las clases anteriores:

- podemos recargar un móvil añadiendo una determinada cantidad

```
public void recargar(int saldo) {  
}
```

- un objeto Punto puede desplazarse a la derecha una distancia determinada

```
public void desplazarDerecha(double distancia) {  
}
```

- una bombilla se puede encender y apagar

```
public void apagarEncender(boolean estado) {  
}
```

- se puede sacar de un almacén una determinada cantidad de un producto

```
public void extraerCantidad(int cantidad) {  
}
```

- la clase Hora nos devuelve la hora actual en formato "hh:mm:ss"

```
public String getHora() {  
}
```

- una bombilla es capaz de indicarnos cuál es su potencia

```
public int getPotencia() {  
}
```