

Programación

Tema 4



Indice

Portada (Pagina 1)

Ejercicios (Pagina 3 – 7)

Conclusión (Pagina 8)

3. Actividades y ejercicios.

1. ¿Qué problemas tiene la programación estructurada?

Es difícil de mantener, depurar, probar y la dificultad de plasmar los conceptos de la vida real.

2. Explicar con tus palabras los fundamentos de la POO.

Abstracción: Poder modelar elementos de la vida real.

Encapsulación: Ocultar trozos de código

Modularización: Dividir elementos para darles a cada uno una función clara y sencilla.

Cohesión: Tener una función clara y sencilla. Tener una alta cohesión

Acoplamiento: Los elementos deben ser los más independientes posibles. Bajo acoplamiento.

Herencia: Que los objetos hereden partes en común.

Polimorfismo: Tener dos o más métodos con el mismo nombre pero que hacen cosas diferentes.

3. ¿Qué es una clase? ¿Y un objeto? ¿Cuántos objetos de una clase se pueden tener?

La clase es un molde para crear objetos.

Un objeto es una clase con un valor inicializado.

Puedes tener todos los objetos que necesites de una misma clase.

4. Las clases unen los datos y los comportamientos en objetos y métodos comunicándose entre ellos usando mensajes.

5. Crear una clase para representar el NIF de una persona.

```
Public class Nif {
```

```
String nombre;  
String apellido1;  
String apellido2;  
int numero_nif;  
char letra_nif  
Boolean sexo;  
String fecha nacimiento;
```

```
String fecha_validez;
```

```
nif1 = new Nif( ){  
}  
public void Nif(){  
}
```

```
public void Nif(String nombre,String apellido1,String apellido2,int numero_nif,char){  
String nombre = new String( nombre )  
String apellido1 = new String( apellido1 )  
String apellido2 = new String( apellido2 )  
this.numero_nif=numero_nif  
}  
public String GetNombre (){  
return nombre  
}  
public void SetNombre(String Nombre){  
this.nombre=nombre  
}  
public int letra{  
letra_nif=numero_nif % 23;  
  
}
```

6. ¿Cuántas clases puede haber en un fichero .java?

Una por fichero.

7. ¿Cómo incluir código externo en ficheros jar en un proyecto sin usar el IDE o gestores de proyecto?

Al hacer el javac usando el parametro -cp o -classpath y la ruta.

8. ¿Qué es un fichero jar?

Un comprimido que tiene clases,metodos y recursos de java.

9. ¿Qué es un constructor? ¿Cuándo se llama?

Es un metodo que se usa para inicializar un objeto.
Se le llama al hacer el new

10. ¿Qué puede suceder si al usar un atributo de una clase se obtiene una excepción de tipo java.lang.NullPointerException?

Que el atributo esta a nulo, es decir que no se ha inicializado

11. ¿Cuántos constructores se puede tener en una clase?

Todos los que quieras mientras los parámetros que piden o que devuelvan sean diferentes.

12. ¿Qué es el constructor por defecto? ¿Qué hace?

Es un constructor que no pide parámetros, sirve para inicializar objetos.

13. El siguiente código tiene al menos 2 fallos importantes, ¿Qué excepciones se causan al ejecutar? ¿A qué es debido?.

El objeto motor no se inicializar, dará null excepcion.

Tampoco se inicializa fecha_revisión, dará null excepcion.

Los atributos no son privados.-

Hay varias clases en el mismo fichero.

```
import java.util.Date;
public class Coche {
    Motor motor;
    String matricula;
    int puertas;
    int peso;

    public void arranca(){
        motor.arrancar();
    }

    public void parar(){
        motor.parar();
    }

    public void imprimir_fecha_revision(){
        System.out.println("La última revisión
        fue"+motor.fecha_revision.toString());
    }

    public static void main(String args[]){
        Coche c;
        c= new Coche();
        c.arranca();
        c.parar();
        c.imprimir_fecha_revision();
    }

    enum ESTADOS_MOTOR{
        ENCENDIDO,
        APAGADO,
        AVERIADO
    }

    class Motor{
```

```
int cilindrada;
int potencia;
Date fecha_revision;
ESTADOS_MOTOR estado;

public void arrancar(){
    estado=ESTADOS_MOTOR.ENCENDIDO;
}

public void parar(){
    estado=ESTADOS_MOTOR.APAGADO;
}
}
```

14. Modificar el código anterior (razonando las modificaciones), para que funcione.

15. Se tiene una librería externa en ./lib para genera pdf llamada JavaPDF.jar. Indicar cómo compilar y ejecutar el programa que se encuentra en ./principal.java y hace uso de la librería.

```
javac .\principal.java -cp ./lib/JavaPDF.jar
```

16. ¿Qué son los modificadores de acceso y que función tienen? ¿A qué elementos se aplica?

Sirven para ocultar parte de código a los que no necesitan verlo,
Se aplica a clases, subclases, métodos y atributos.

18. ¿Qué es this? ¿Para qué se utiliza this?

Se utiliza para cuando haces referencia a atributos dentro de su misma clase para darle más legibilidad al código.

19. En Java los parámetros se pasan por valor, pero los cambios en los objetos dentro de métodos son permanentes. Explicar la razón.

Los tipos de datos primitivos se pasan por valor pero los objetos se pasan por referencia.

20. Indicar la diferencia entre usar == y el método equals al comparar objetos.

== compara la dirección de memoria mientras que equals compara el contenido.

21. ¿Qué es un atributo estático? ¿Para qué sirve?

Es un atributo a nivel de clase que no cambia su valor.

22. Se tiene un juego de estrategia y se quiere contar el número de arqueros totales creado. ¿Cómo hacerlo?

Con un contador que cada vez que se ejecute un constructor aumente su valor.

23. ¿Es recomendable declarar los atributos como privados? De ser así ¿Cómo se accede y modifica dichos atributos?

Si, usando getters y setters

24. ¿Qué es una propiedad? ¿Existe en Java?.

En java existen los atributos que son como las propiedades.

25. En C++ ¿Como se destruyen los objetos? ¿Es necesario desarrollar algo en la clase para poder destruirlos?

Liberando el espacio que ocupan en memoria usando el operador ~

26. ¿Cómo se destruyen los objetos en Java?

El recolector de basura lo elimina cuando no se utiliza

27. ¿Cómo sabe el GC que puede liberar el espacio reservado para un objeto?

Apuntando un objeto a null

28. Definir qué es un paquete.

Un conjunto de clases y atributos con una relación lógica

29. Explicar la forma de desarrollar paquetes con un algo grado de cohesión y bajo acoplamiento.

Agrupándolos siguiendo una relación lógica

30. ¿Qué es un fichero JAR? ¿Qué contiene?

Es un fichero comprimido que tiene todos los ficheros necesarios como paquetes y librerías externas de un proyecto

31. Se quiere distribuir una aplicación y asegurarse de que funcionará incluso sin Java instalado en los equipos. ¿Cómo hacerlo?

Usando modulos

32. ¿Qué herramienta se utiliza la documentación en Java? ¿Qué son los tags?

Javadocs, los tags son comentarios que utiliza javadoc para crear la documentación.

Conclusión					
Dificultad	☆	☆	☆	☆	☆
Interesante	☆	☆	☆	☆	☆
Rápido	☆	☆	☆	☆	☆

