**Ejercicio**

Intenta realizar una clase, de nombre Examen, para guardar información sobre los exámenes de un centro educativo. La información que se guarda de un examen es la siguiente: el nombre de la asignatura, el aula, la fecha y la hora. Para guardar la fecha y la hora, hay que realizar dos clases, Fecha y Hora.

La clase Fecha guarda día, mes y año. Todos los valores se reciben en el constructor por parámetro. Además, esta clase debe tener un método que devuelva cada uno de los atributos, y un método toString() que devuelva la información de la fecha en forma de String.

La clase Hora guarda hora y minuto. También recibe los valores para los atributos por parámetro en el constructor, tiene métodos que devuelven cada uno de los atributos y un método toString().

**Como puede hacerse**

La clase Fecha tiene tres atributos de tipo entero, cuyos valores se reciben en el constructor por parámetro. El método toString() devuelve los valores de los tres atributos separados por una barra («/»). La clase hora tiene dos atributos de tipo entero, cuyos valores se reciben en el constructor por parámetro. El método toString() devuelve los atributos separados por puntos(“: “).

La clase Examen realiza una composición de elementos de tipo String, Fecha y Hora. Recibe los valores de los atributos por parámetro en el constructor, y además tiene un método para cambiar cada uno de los atributos y un método para devolver cada uno de los atributos. También tiene un método toString() para devolver el valor de los atributos en forma de String.

PREGUNTAS SOBRE POO

**¿Cuál es la diferencia entre una Clase y un Objeto?**

La clase es un tipo de plantilla con codigo capaz de crear múltiples objetos. Para poder crear un objeto necesito haber creado una clase anteriormente con sus atributos y métodos para poder instanciar ese objeto. Por lo tanto, la clase es donde configuramos nuestro codigo para después reutilizarlo para poder crear un Objeto.

**¿Qué es la Herencia?**

**La herencia la utilizamos para reutilizar codigo. Gracias a esta posibilidad de poder instanciar objetos sin tener que crear de nuevo un codigo, si no utilizamos la herencia para que ese objeto tengo unos atributos y métodos predefinidos, utilizando la palabra reservada Extends en la subclase o clase hija.**

**¿Qué es la Encapsulación? ¿Qué niveles hay?**

La encapsulación se utiliza para ocultar los detalles del atributo o método. Utilizamos la encapsulación con la palabra reservada Private para que asi no modifiquemos el valor de ese atributo o método si no que creamos una copia del mismo. Para poder acceder a estos métodos o atributos utilizamos los métodos setters y getters, asi pudiendo establecer o acceder a estos miembros de la clase que son private.

**¿Qué es Polimorfismo?**

**El polimorfismo significa poli= mucho y morfismo = forma, es decir, muchas formas. Es la capacidad que tiene el lenguaje de utilizar un objeto como molde y poder moldear otro objeto. Lo que hacemos es usar un molde especifico para nuestro objeto, pero después le damos la forma que necesitemos, es decir, el tipo de variable por ejemplo Coche coche, pero en el constructor usamos el es = new furgoneta () o = new deportivo (). En este ejemplo estamos usando polimorfismo, ya que estamos creando la variable de tipo Coche, pero en el constructor finalmente utilizamos la forma definitiva de nuestro objeto. Aquí también usamos la herencia ya que heredamos esa forma principal, es decir, tendrá puertas, espejos, etc.**

**¿Qué es Sobrecarga?**

**Existe la sobrecarga de constructores y métodos. Es decir, podemos llamar a un método de la misma forma siempre y cuando se diferencia en los tipos o cantidades en los parámetros. Ejemplo:**

* **Void sumar (int num1, int num2) {…}**
* **Void sumar (int num1, int num2, int num3) {…}**

**Esto es una sobrecarga de un método lo llamamos igual, pero al invocarlo tenemos dos métodos llamados sumar que se diferencia por la cantidad de parámetros y no por su tipo, pero también cabe la posibilidad de que sea por tipo:**

* **Void sumar (int num1, int num2) {…}**
* **Void sumar (double num1, double num2) {…}**

**Diferencia entre Interfaz y Clase *(\*\*esta la veremos más en detalle en clase, pero buscas información para ir teniendo claro los conceptos)***

***Una Clase tiene una estructura con métodos y atributos mientras que la Interfaz lo que contiene son métodos que deben ser implementados en aquellas clases que la implemente con la palabra reservada, implements.***

***Una clase se usa para crear instancias de objetos, mientras que una interfaz se utiliza para definir comportamientos que pueden ser compartidos por diferentes clases.***

***Una clase puede tener atributos con visibilidad privada, constructores y métodos con o sin implementación. Una interfaz define un conjunto de métodos (abstractos o con implementación) que deben ser implementados por las clases que la utilizan.***

**¿Pueden las clases en Java heredar más de una clase? *(\*\*esto sí lo he contado en clase, va de la mano de la anterior pero algo os he contado)***

**En Java no existe, si en otros lenguajes como C, la herencia múltiple. Por eso las Interfaces nos ayudan mucho en este aspecto.**

**¿Qué significa “Paso por valor” y “Paso por referencia”?**

**El paso por valor es cuando en la invocación a un método que tiene parámetros, en la llamada es argumento, tu tienes que meter directamente en el argumento un valor.**

**En el paso por referencia lo que hacemos en el argumento le pasamos una dirección de memoria.**

**¿Qué hace la palabra Static?**

**La palabra clave Static se utiliza para indicar al programa que un método o atributo (variable) pertenece a la Clase y no a un objeto. Es decir, puedo utilizar tanto métodos como variables estáticas sin la necesidad de instanciar ningún objeto. También se pueden crear clases estáticas.**

**¿Qué son las Excepciones?**

**Las Excepciones es un evento o condición anormal que ocurre durante la ejecución de un programa y que interrumpe el flujo normal de instrucciones. Estos errores en ejecución tenemos la posibilidad de capturarlos previamente en el codigo.**

* **Existen varios tipos de excepciones diferentes:**

1. **Errores en tiempo de ejecución.**
2. **Errores lógicos.**
3. **Condiciones impredecibles.**

**Realmente las Excepciones es algo que hay que estudiar con mas detalle ya que es mas complejo que todo esto.**

**¿Qué tipo de excepciones conoces?¿Con que sentencia se maneja las excepciones?**

***Excepciones No Verificadas (Unchecked Exceptions):***

**No requieren que el programador las maneje explícitamente.**

**Son subclases de RuntimeException y generalmente se relacionan con errores de programación o condiciones impredecibles.**

**Ejemplos de excepciones no verificadas:**

***ArithmeticException:* Se lanza cuando ocurre una operación aritmética inválida (por ejemplo, división por cero).**

***NullPointerException:* Se lanza cuando se intenta acceder a un objeto que es null.**

***ArrayIndexOutOfBoundsException:* Se lanza cuando se intenta acceder a un índice fuera del rango de un array.**

***NumberFormatException:* Se lanza cuando se intenta convertir una cadena a un número, pero la cadena no tiene el formato correcto.**

***Excepciones Verificadas (Checked Exceptions):***

**Requieren que el programador las maneje explícitamente (o declare que el método las arrojará).**

**Se derivan de la clase Exception (pero no de RuntimeException).**

**Ejemplos de excepciones verificadas:**

***IOException:* Se lanza cuando hay problemas con la entrada/salida, como la lectura o escritura de archivos.**

***ClassNotFoundException:* Se lanza cuando se intenta cargar una clase que no se encuentra.**

***SQLException:* Se lanza cuando ocurre un error en una operación de base de datos.**