**3.14 ANEXO C**

**–**

**CUESTIONES**

1.¿Qué es mejor, almacenar la fecha de nacimiento o la edad? Razona tu respuesta

La fecha de nacimiento ya que siempre que tengamos un atributo que pueda ser calculado con un método deberíamos almacenar los datos y calcularlo, además en el caso de la edad es muy obvio ya que las fechas van cambiando.

2.¿Cómo se implementan las operaciones de una clase?

A través de los métodos.

3.¿Puede una clase tener varios constructores? ¿Para qué?

Si porque, puede recibir varios parámetros o ninguno etc… la función de tener varios constructores tener diferentes formas de inicializar un objeto.

4.¿Es correcta la expresión:

**((b>4) \* (c<2)**? Razona tu respuesta

Es incorrecta porque estas multiplicando dos booleanos.

5.Escribe la signatura de 3 métodos que formen parte de la interface de la clase String. Consulta la documentación en la API de Java en Oracle.

charAt que devuelve un char de donde este la posición del parámetro que le pases.

valueOf que convierte lo que le pases a String.

length que te devuelve la longitud de la cadena.

6.¿Puede un accesor no recibir ningún parámetro? Pon un ejemplo y razona tu respuesta

Si, como norma general los accesores que nosotros creamos y que llamamos gets no se les pasa parámetro.

getAlias():String

7.¿Puede un mutador no recibir ningún parámetro? Pon un ejemplo y razona tu respuesta

Aunque por norma general a un mutador le debemos pasar parámetro, podemos crear mutadores que modifiquen el estado de un objeto de forma automática, como por ejemplo en el ejemplo del Robot que modificamos su orientación de forma automática.

8.¿Qué pasaría si no cumpliéramos la ENCASULACION y OCULTACION de datos usando nivel de acceso **public** para los atributos de una clase?

Que podríamos modificar los estados de los objetos desde cualquier parte. Y podríamos violar restricciones.

9.¿Cuál es el ámbito de una variable de clase? ¿Y su tiempo de vida? Razona la respuesta

Desde la línea donde está declarada hasta el final del cuerpo de la clase. Hasta que muera el objeto.

10.¿Para qué serviría definir una variable local estática ( **static**)? Haz una prueba y razona

tu respuesta. Idea: comprueba si la variable tiene un tiempo de vida distinto al que

esperas...

las variables locales staticas en java no existen.

Pero en otros lenguajes si están permitidos.

Sirve para retener el valor de la variable dentro de las llamadas al método, con el objetivo de tener guardado ese valor para las futuras llamadas.

11.¿Puede un método de clase ser privado? Haz un ejemplo y razona tu respuesta.

Si, porque a lo mejor ese método no te interesa que forma parte de la interfaz de la clase, o a lo mejor ese método sirve como método auxilar para otro método.

12.¿Cómo se pueden controlar las restricciones sobre atributos?

En el mutador, cada vez que un atributo tenga una restricción debemos crear un mutador y controlarlas ahí.

13.Piensa e implementa un mutador que permita modificar un atributo de tipo String (supongamos que a nivel de diseño se ha establecido una restricción NOT NULL y otra NO VACIO sobre el atributo)

Public void setAlias(String alias){

Assert alias.lenght >=1 : “Error…..”;

Assert alias != Null : “Error…”;

This.alias = alias;

}

14.¿Es posible tener dos métodos en una clase cuyas signaturas sean iguales excepto el tipo de retorno? ¿Es un caso de sobrecarga?

Además tiene que variar los parámetros que le pasemos al método para poder sobrecargarlos.

15.¿Por qué el uso de aserciones no es la mejor solución para el control de restricciones en mutadores?

Porque las aserciones paran la ejecución del programa impidiendo así que se puede solucionar el error.