1. Dada la siguiente clase

public class Persona{

private String nombre;

private String apellido;

public Persona(String nombre, String apellido){

this.nombre = nombre;

this.apellido = apellido;

}

public String getNombre(){

return this.nombre;

}

public void setNombre(String nombre){

this.nombre = nombre;

}

public String getApellido(){

return this.apellido;

}

public void setApellido(String apellido){

this.apellido = apellido;

}

}

* 1. Donde se observa el encapsulamiento?

**Rpta:** Se observa en que el modificador de acceso de los atributos de la clase persona, están private y en que los de los métodos están en public. Esto hace que los atributos solo sean accesibles desde los métodos establecidos.

* 1. En el constructor, identificar que es una variable y que es un atributo

**Rpta:** En el constructor, this.nombre y this.apellido hace referencia a los atributos de la clase, mientras que nombre y apellido son variables locales.

1. ¿Qué mensajes debería poder enviarle un objeto de la clase Docente a un objeto de la clase Alumno? ¿Y viceversa? Ejemplificar cómo se enviarían dichos mensajes.

**Rpta:** Debería poder enviarle la nota del primer parcial, del segundo, y si desaprueba, promociona o va a final. El alumno debería poder enviarle, parcial 1 , parcial 2, y en caso de recuperatorio y/o final, los mismos.

Clase docente:

Alumno.entregarParcial(1);

Alumno.setNota();

Alumno.asistirAClase();

Clase alumno:

Docente.responderConsulta(String);

Docente.solicitarRevisionParcial(1);

Public void solicitarRevision(){

Docente.revisarParcial(1);

}

Public class Docentes{

public void corregirTP(){

while(){

Alumno alumno = new Alumno();

Alumno.entregarTrabajoPractico();

Alumno.setNota();

}

}

1. Se desea modelar el funcionamiento de un cajero automático. El mismo debe estar capacitado para realizar las siguientes operaciones
   1. Depositar dinero
   2. Extraer dinero
   3. Consultar el saldo

Una vez obtenida la clase que responda a este funcionamiento, se solicita testear la misma de la siguiente manera:

* 1. Crear una cuenta vacía
  2. Crear una cuenta con saldo inicial
  3. Ingresar dinero a la cuenta
  4. Retirar dinero de la cuenta
  5. Consultar el saldo de la cuenta

1. ¿Cuál será el resultado de ejecutar el siguiente caso de prueba de JUnit? ¿Por Qué?

package ar.edu.unlam.basica2;

import static org.junit.Assert.\*;

import org.junit.Test;

public class TestString {

@Test

public void testIgualdadDeObjetos(){

String cadena1 = new String("Cadena");

String cadena2 = new String("Cadena");

assertTrue(cadena1 == cadena2);

}

}

**Rpta:** Da fail porque estamos comparando referencias a objetos (direcciones de memoria). Si hubiéramos usado assertEquals(cadena1, cadena2); si daba verdadero porque compara contenido. Se usa asi:

if(cadena1.equals(cadena2)){

}

1. Un método estático, ¿puede referenciar un atributo no estático de la clase? ¿Por Qué?
2. Dos de los siguientes tipos no son primitivos del lenguaje. ¿Cuales?

int, double, Float, boolean, byte, String, char

1. ¿Porqué es necesario instanciar una clase para poder utilizar sus miembros no estáticos?
2. ¿Que valor por defecto toma una variable de un tipo no primitivo con sólo declararla?

Actor alfredoAlcon;

1. A qué método se invoca cuando utilizamos la palabra reservada new? ¿Qué se obtiene como resultado?

alfredoAlcon = new Actor();

1. ¿Con qué valores por default se creará cada posición del siguiente array?

Integer vectorDeNumeros = new vectorDeNumeros[5];