

PROTOCOLO PARA SITUACIONES DE EXAMEN

Estimados Docentes:

A los efectos de optimizar el trabajo de todos les recordamos tener presente lo siguiente:

1. INTEGRACIÓN DEL TRIBUNAL:

“La presidencia será ejercida por el profesor de mayor grado en el Escalafón Docente y este grado el de mayor antigüedad.”
(Art. 79 del REPAG)

2. CASOS DE ALUMNOS CON PLAN DE INTERVENCIÓN MULTIDISCIPLINARIA (TOLERANCIA):

Se les recuerda que al evaluar a los alumnos deben tener en cuenta sus necesidades educativas especiales. Esto es: prueba escrita adecuada a las dificultades de alumno, otorgarle mayor tiempo, explicarle la consigna en forma personalizada, aclaración de algún dato de la propuesta que no entienda, priorizar la prueba oral.

3. FALLOS DE LOS TRIBUNALES EXAMINADORES (Artículo 93 del REPAG)

En caso de aprobación el fallo será “aprobado con la calificación de suficiente que corresponda”.
En el caso de no aprobación, el fallo será “no aprobado con la calificación de insuficiencia que corresponda”. (NO corresponde “eliminado”)

Artículo 73.-

Para la aprobación de un examen Reglamentado se deberá tener en cuenta el siguiente procedimiento:

- 1) En exámenes correspondientes a Asignaturas con régimen de exoneración, la primera instancia del examen será eliminatoria (con calificación menor o igual al 6).
Pasarán a la segunda instancia aquellos alumnos que obtengan calificación mayor o igual a 7 en la primera. La calificación final del examen se obtendrá del promedio entre las calificaciones de la primera y la segunda instancia

ESCRITO	ORAL	FALLO
< 7	NO CORRESPONDE	NO APROBADO CON....
≥ 7	CORRESPONDE	(SE PROMEDIAN AMBAS CALIFICACIONES): A) APROBADO CON.... B) NO APROBADO CON....

En exámenes correspondientes a Asignaturas con régimen de examen obligatorio (Matemática de 2° y 3° y el ECT de 3°) podrán distinguirse siguientes situaciones:

NOTA DEL AÑO	NOTA ESCRITO	NOTA ORAL	FALLO
< 7	< 7	NO CORRESPONDE	NO APROBADO CON....
< 7	≥ 7	< 7	NO APROBADO CON....
< 7	≥ 7	≥ 7	APROBADO CON....
≥ 7	1,2,3	NO CORRESPONDE	NO APROBADO CON....
≥ 7	4,5,6	< 7	NO APROBADO CON....
≥ 7	4,5,6	≥ 7	APROBADO CON....
≥ 7	≥ 7	NO CORRESPONDE	APROBADO CON....

Artículo 74.-

En un examen Libre, la calificación final se obtiene mediante el promedio de la primera y la segunda instancia, ambas eliminatorias (con calificaciones menores que 7) Para su aprobación será necesaria la calificación 7 (o superior).

4. ESCRITURADO Y PRESENTACIÓN:

- a. las Actas deberán escriturarse con letra clara y prolija, sin ningún tipo de tachadura o enmienda.
- b. Al finalizar sus actuaciones el Tribunal Examinador entregará una copia de la propuesta de examen junto con las actas correspondientes.

I.S.B.O. - Examen de Programación II

Nombre y Apellido: Diego Lopez C.I.: 12345678

1. Construir una ventana que contenga un botón ACEPTAR. Al hacer clic en el botón aparecerá una nueva ventana con el siguiente mensaje: "Examen de Programación II".
2. Un alumno está identificado por nombre, sexo y edad. El nombre será visto desde todas las clases en tanto que la edad y el sexo sólo desde su clase. Construir la clase Alumno con dos constructores distintos y los siguientes métodos: get, set y esMayor. El último método retornará true o false dependiendo de si el alumno es mayor de edad o no.
3. Construir un ArrayList de tipo String y 4 métodos de acceso.

1. Completar con la palabra correspondiente los espacios en blanco de la siguiente frase:

Los métodos son funciones de una clase. Generalmente los métodos se dividen en aquellos que se usan internamente en la clase, llamados métodos, los que se usan fuera de la clase, llamados métodos y los que son usados por la clase y sus derivadas, llamados métodos

2. En un proyecto examen crear un paquete donde se guardará la clase persona cuyos atributos son cédula, nombre y apellido. En su correspondiente capa crear una ventana que permita solicitar al usuario que ingrese los datos de uno o más personas.
3. Mejorar el código desarrollado en la pregunta 2 para que se pueda trabajar en memoria con los datos de más de una persona. Y mostrarlos en pantalla

```
lista.add(new Persona("maria",29));
    lista.add(new Persona("Alina",48));
    lista.add(new Persona("mia",1));
    lista.add(new Persona("cuatro",1));
```

5 que hace el siguiente método getFigura

```
public static Figura getFigura(int i){
    Figura f=null;
    switch (i){
        case 1: f=new Cuadrado();
        case 2 : f=new Circulo();
    }
    return f;
}
```

6 - Código para crear una ventana
dar un ejemplo

1 – Explicar los cambios que se realizan: `chmod 641 archivito.txt`

2 – Políticas de respaldo

3 – Explicar que entiende por `init 5`

4 - respaldar el directorio `/etc` y luego recuperarlo en el directorio `/home`

RESPUESTA:

1-

Lectura tiene el valor de 4

Escritura tiene el valor de 2

Ejecución tiene el valor de 1

Por lo tanto el archivito.txt quedaría con permisos de lectura y escritura para el dueño,

Permisos de lectura para el grupo, permisos de ejecución para otros.

2 – *total, diferencial, incrementa*

3-*Init 5 permite al sistema iniciar en un nivel con servicios multiusuarios completos, redes, etc. Es el modo de inicio que está instalado por defecto.*

```
4-tar -cvf backup.tar /etc
   Cd /home
   tar -xvf /dondeseencuentre/backup.tar
```

EXAMEN DE PROGRAMACIÓN II – SETIEMBRE 2012

NOMBRE:.....

EJERCICIO 1:

A partir de la siguiente función llamada “funcionvec”, ¿cómo quedaría el vector que está a continuación: {21, 23, 20, 25, 22}

```
void funcionvec(int vector[], int tamaño)
{
    int pos, num, i, temp;

    pos = 0;
    num = vector[pos];
    for (i=1; i<tamaño; i++)
    {
        if (num < vector[i])
        {
            pos = i;
            num = vector[i];
        }
    }
    vector[pos] = vector[0];
    vector[0] = num;
}
```

EJERCICIO 2:

Diseñar un programa que intercambie el valor de dos números enteros. Por ejemplo: Si **a** y **b** valen 1 y 2 respectivamente, la llamada a la función **intercambia(a,b)** hará que **a** pase a valer 2 y **b** pase a valer 1.

EJERCICIO 3:

Escribir un programa que presente el siguiente menú:

1 – Impares

Aquí se deberá mostrar en pantalla los números impares entre 0 y 100.

2 – Tablas de multiplicar

En esta opción, se le tiene que pedir al usuario la tabla y se debe desplegar en pantalla la tabla correspondiente.

3 – Menor, mayor

Se debe ingresar un número y mostrar en pantalla según lo que corresponda:

- Es menor a 10
- Está entre 10 y 100
- Es mayor a 100

Se debe implementar cada caso con funciones.

EXAMEN PROGRAMACIÓN II – PERÍODO DICIEMBRE 2012

NOMBRE:..... FECHA:

1 – Explique la diferencia entre variables globales y variables locales.

2 - ¿Qué diferencia hay entre un while y do-while?

3 - ¿Cómo se declara un vector que contenga 15 números enteros.

4 - ¿Cuál de las dos sentencias que se describen a continuación es una comparación entre dos variables enteras? 1) `a == b;` 2) `a = b;`

5 - En un vector que contiene una cadena de 15 caracteres, ¿cómo se declara y cuál es el último carácter?

6a- ¿Para qué se utilizan `fwrite()` y `fread()` en los archivos binarios y cuál es su sintaxis?

6b - ¿Para qué se utilizan `fprintf()` y `fscanf()` en los archivos de texto y cuál es su sintaxis?

7 - Si en una parte del programa tenemos el siguiente código:

```
a = 3;
for (i=1; i<= 3 ; i++)
{
    printf("%d\t", a+i);
}
getch();
```

¿Qué aparece en pantalla?

8 - Definir un vector de tipo estructura de 500 elementos, en cada estructura se almacenan los siguientes datos de empleados:

- cédula del empleado
- nombre del empleado
- edad del empleado
- domicilio del empleado

9 – Escribir un programa que presente el siguiente menú:

- 1 – Cargar un vector de números enteros
- 2 – Mostrar el vector
- 3 – Buscar un número dentro del vector
- 4 – Salir

Implementar cada una de las opciones del menú mediante funciones.

Nombre y apellido	Cédula de Identidad

Programación II

Exámen Febrero 2013

1) Dados los siguientes datos de clientes:

- N° de cliente
- Nombre
- Apellido
- Teléfono

Escribir un programa que defina la estructura, y que mediante un menú permita guardar los datos en un archivo binario (verificando si existe el numero de cliente), así como listarlos.

2) Definir y declarar un puntero.

3) Explicar los distintos modos de apertura de fopen.

4) Definir la sintaxis de fwrite, fgets, fprintf y fread.

5) Declare un string de 20 caracteres y cargue un valor.

6) Indique para que sirve el operador * al trabajar con punteros

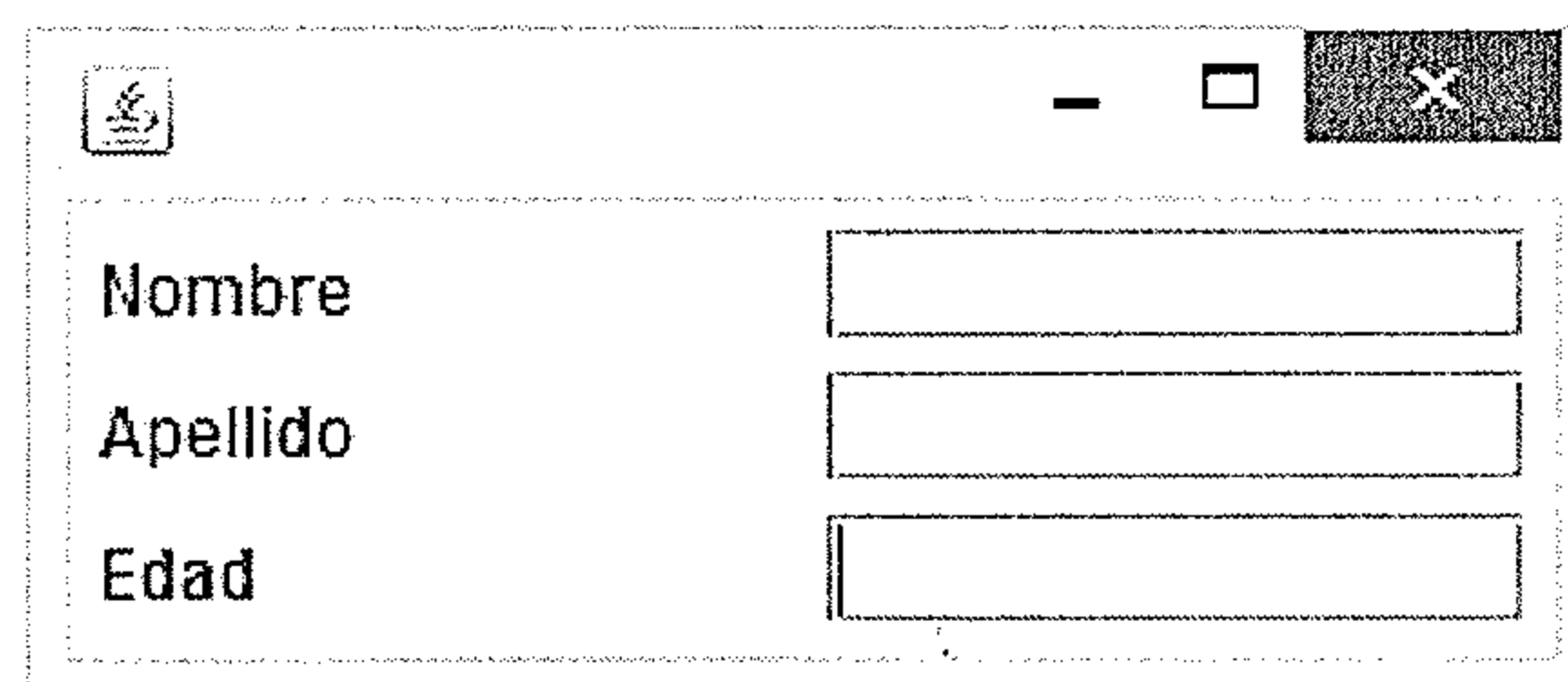
7) Defina un array entero de 20 posiciones, y elimine la basura en él.

Programacion II Febrero 2014

Nombre: _____
C.I.: _____
Cantidad de hojas: _____

Se tiene la clase PanelDatos la cual cuenta con un formulario para ingresar datos como se puede ver en la siguiente imagen.

Ventana utilizando un PanelDatos sin mas.

A screenshot of a Java Swing window titled "PanelDatos". The window has a standard title bar with a minimize button, a maximize button, and a close button. Inside the window, there is a form with three text input fields. The first field is labeled "Nombre", the second is labeled "Apellido", and the third is labeled "Edad".

La clase cuenta con las siguientes funciones:

String getNombre() Devuelve el valor del primer campo.

String getApellido() Devuelve el valor del segundo campo.

String getEdad() Devuelve el valor del tercer campo.

void borrar() Elimina el contenido de los 3 campos.

void setApellido(String Ape) Cambia el contenido del segundo campo al contenido en Ape.

1) Se pide implementar la clase DatosFamilia, la cual disponga de dos formularios con el formato ya mostrado que permitan ingresar información de un Padre y un Hijo. Se pide también que para cada formulario se cuente con un botón de borrado, el cual al ser presionado elimine el contenido de dicho formulario. Por último se debe de contar con un botón el cual junte la información obtenida por ambos formularios y la despliegue (de forma coherente) en donde se quiera (un texto, un mensaje, ...).

2) Debido a la gran cantidad de familias con el apellido "Gonzalez" se considera útil que al borrar un formulario este mantenga en el segundo campo el valor "Gonzalez". Cree una clase PanelDatos2 que extienda a PanelDatos el cual en caso de suplir al original en DatosFamilia cambie el comportamiento haciendo que los formularios mantengan un "Gonzalez" en el segundo campo.

2010
11/2
~~11~~

Examen - Programación X - ISBO - 11/2/2014

Nombre: _____

El punto 1 es obligatorio y debe optar por 3 de los 4 restantes.

Cada uno de los ejercicios tiene un valor de 3 puntos.

Ejercicios:

1. Cree una ~~clase~~ llamada Goles Jugadores para almacenar (en un arreglo) los goles convertidos hasta el momento por los jugadores titulares de un equipo de futbol (del n° 1 al n° 11) que implemente un menú con la las siguientes opciones:

1. **Cargar goles de los jugadores.**
 2. **Mostrar los goles de todos los jugadores.**
 3. **Mostar mayor cantidad de goles convertidos por un jugador.**
 4. **Mostar el promedio de los goles convertidos.**
 0. **Salir.**
- No corre ok*

Las opciones del menú llaman a métodos que no debe implementar en este punto.

2. Crear un método para cargar por consola los goles de los jugadores (ítem 1 del menú).
3. Crear un método que muestre por pantalla el número de jugador y los goles convertidos, solo debe mostrar los jugadores que tengan goles. (ítem 2 del menú).
4. Crear un método que devuelva o muestre la mayor cantidad de goles convertida por un jugador (ítem 3 del menú).
5. Crear un método que devuelva o muestre el promedio de goles convertidos (ítem 4 del menú).

FEBRERO 2016

Examen programación II

Ejercicio 1

Hacer una aplicación que desde una clase principal instancie una clase que muestre una ventana

Ejercicio 2

Desde una clase llamar a atributos y métodos que estén en otra clase (el alumno codificará las 2 clases)

Ejercicio 3

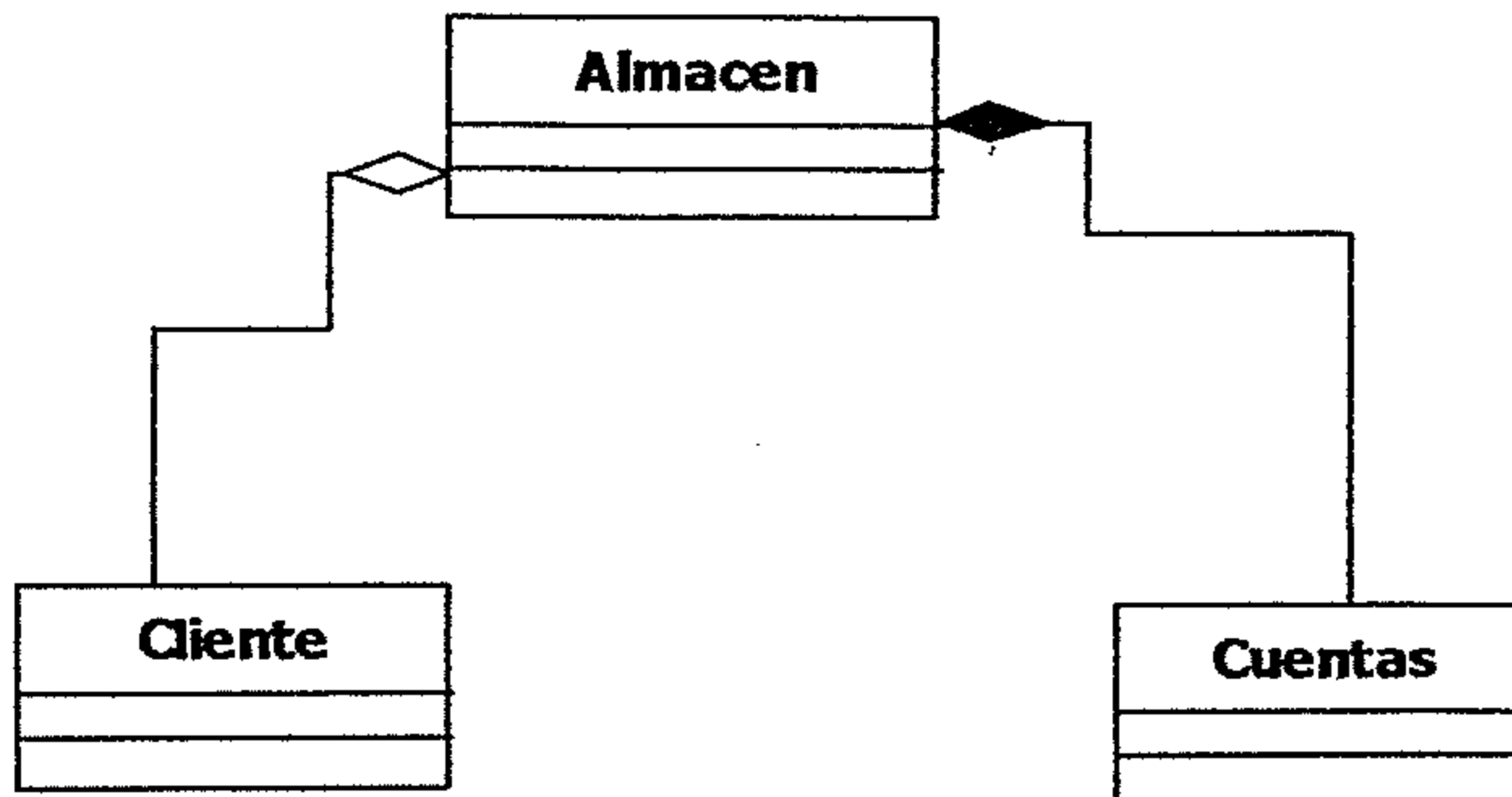
a) ¿Cuáles son las principales características de la programación orientada a objetos? Explique y ejemplifique cada una de ellas.

Ejercicio 4

a) Que hace la siguiente clase

```
public class Profesor extends Persona {  
  
    private String materia; // propiedad  
    private String cargo; // propiedad  
  
    public Profesor () { // constructor  
    }  
    public String getCargo () { // asesor de cargo  
        return cargo;  
    }  
    public void setCargo (String val) { // mutador de cargo  
        this.cargo = val;  
    }  
    public String getMateria () { // asesor de materia  
        return materia;  
    }  
    public void setMateria (String val) { // mutador de materia  
        this.materia = val;  
    }  
}
```

- 1- Explique y desarrolle como se realiza la programación en capas.
- 2- Definición de :
 - a. Herencia
 - b. Polimorfismo
 - c. P.O.O.
- 3- Observe el siguiente diagrama de clase y explique las relaciones que allí se expresan.



- 4- La veterinaria "El bicho" lleva un registro de sus pacientes. Todos los animales tienen nombre y cantidad de patas. En el caso de los perros se conoce la raza. En el caso de las aves nombre de la especie y color del plumaje.
 - a. Desarrollar diagrama de clases En UML
 - b. Pasar al lenguaje de programación java
 - c. Crear un método que permita reproducir el sonido de cada animal (modo texto).
- 5- Dado el siguiente código:
 - a. Analice y explique acciones que se realizan.
 - b. Crear un objeto de la Clase Persona y cargar datos

```

final JLabel lblCedula = new JLabel("CEDULA");
contentPane.add(lblCedula, "cell 2 0");

final JLabel lblNombre = new JLabel("NOMBRE");
contentPane.add(lblNombre, "cell 2 1");

final JLabel lblApellido = new JLabel("APELLIDO\r\n");
contentPane.add(lblApellido, "cell 2 2");

```

```

JButton btnAlta = new JButton("New button");
btnAlta.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

    }
});
contentPane.add(btnAlta, "cell 2 3");

```

Programación II

ProgramacionFebrero2015.txt

Opcion 1

Dada la siguiente clase

```
public class Botones extends JFrame {
    public JButton [] botones;
    public Botones(int i){
        botones = new JButton[i] ;
        setLayout(new FlowLayout());
        for (int j=0;j<i;j++){
            botones[j]=new JButton("boton"+(j+1));
            add( botones[j]);
        }
    }
}
```

Hacer una aplicación gráfica que usando la clase anterior muestre 4 botones y que cuando se aprete un botón muestre cuantas veces se apreto ese botón

Opcion 2

1 - código necesario para usar un campo estático a1 de una clase A (1 punto)

2 - código en java para usar un campo private m1 de una clase A desde una clase B (1 punto)

3 - Codificar un método que devuelva un array de enteros de tamaño 3 (2 puntos)

4 - Dada la siguiente clase Alumno,

```
public class Alumno {
    private String nombre;
    private int edad;
}
```

A - Agregar en la clase Alumno los métodos setters y getters correspondientes (1 punto)

B - realizar una clase Principal que tenga el método main, donde se defina una variable de nombre miLista, miLista es de tipo ArrayList, miLista debe almacenar objetos de la clase Alumno. Hacer un método que almacene 4 alumnos distintos en miLista (2 puntos)

5 - dar un ejemplo de polimorfismo en JAVA y explicarlo (2 punto)

6 - Para que se usen las interfaces en java (1 punto)

7 - Dada la siguiente clase Alumno,

```
public class Alumno {
    private String nombre;
    public int edad;
```

}
Representarla en UML (1 punto)

8 - que devuelve el método en la siguiente instrucción
String entero = config.get("initial"); (1 punto)

Nombre:

Programación II
ESI
Marzo 2015

1) Se comenzó a crear el siguiente diagrama conceptual de clases:

Aviones
-numero -modelo -capacidad

a) Incorporar aviones **civiles y aviones presidenciales**.

De los aviones **civiles** interesa saber: la empresa importadora, si admiten vuelos internacionales o nacionales, tipo de motor que tienen (Mecánico o eléctrico).

De los aviones **presidenciales** interesa saber: a qué país pertenecen, cantidad de pisos, empresa de seguridad que instaló los sistemas de protección.

Aviones_Militares
-Peso -Año -Subtategoría -Cant_motores

Los **aviones militares** son piloteados por **pilotos** que cuentan con experiencia en batallas, se les incluye en la categoría llamada A y esta se identifica también por pilotos solo hombres. Los **aviones presidenciales o civiles** son piloteados por pilotos de cualquier género y los que realizan este trabajo están en la categoría B. De todo piloto se sabe; su nombre, su

identificación, edad, estado civil, horas de vuelo.

Indique los atributos que considere apropiados para esta realidad. Los pilotos que manejan aviones militares tienen una única nave asignada. A su vez cada avión militar contará con dos pilotos experimentados para cada vuelo.

Los demás aviones (civiles y presidenciales) tienen pilotos asignados al azar, dato que no interesa registrar.

b) Complete el diagrama conceptual agregado lo referido a la representación de pilotos y como se relaciona con la representación de aviones. No indague en métodos !

c) Donde se está garantizando el uso de herencia y encapsulamiento? Justifique.

2) Observe este código referente en diagrama del ejercicio 1

a) Comente, donde se indica con * (asterisco) que sucede en esa línea

b) Según la realidad planteada y viendo la construcción de la sub clase Aviones_Militares, se nota algún error y/u omisión. Identifique y corrija.

c) Complete el código para construir la clase AMilitar correctamente. Ingrese lo que falta.

d) Los atributos de la clase Avion (número, modelo y capacidad) que definen en dicha clase, que modificador de alcance deben tener para poder ser usados por la subclase? por qué?

```

public class Aviones_Militares extends Aviones {
    private double peso;
    private int cantMotores;

    public Aviones_Militares(double peso, int cantMotores){ * _____
        super(numero, modelo, capacidad); * _____
        peso=p;
        cantMotores=Cmot;
    }

    public double getpeso() {
        return peso;
    }
    public void setpeso(String peso) {
        this.peso = peso;
    }
}

```

3) Hacer una clase que permita cargar el nombre y la raza de un perro. Luego mostrarlos en pantalla. Ingrese datos por teclado y use los métodos: cargarDatos y mostrarDatos para lo solicitado.

EXM DICIEMBRE 2015

EXAMEN PROGRAMACION 2

Ejercicio 1- Sea la clase VideoJuego, con atributos nombre (String), cantidad de niveles (entero) y año de creación (entero).

Se pide: definir un constructor por defecto que inicialice nivel en 1 y el año en 2012. El nombre por defecto del videojuego es "Sin definir".

Y un constructor por parámetro (con todos los atributos) inicializado de la misma manera. Definir métodos de acceso y modificación (get's y set's).

Ejercicio 2 - Dada la clase Coche:

```
public class Coche {  
    private String matricula;  
    private String marca;  
    private String modelo;  
    private int Km;  
}
```

Nota: Se asume que cada atributo cuenta con métodos de acceso y modificación.
Se pide:

- Anotar la definición de un ArrayList que contenga información sobre coche.
- Realizar un método **agregarCoche(String miMatricula, String laMarca, String elModelo, int losKm)** que permite agregar un coche en la lista creada anteriormente.

Ejercicio 3 -

Basándose en la clase Coche del ejercicio ²2, realizar una pantalla (Swing) que solicite la entrada de los datos, que contenga 2 botones (aceptar y cancelar) cada uno deberá tener la programación correspondiente a la acción del botón.

Nombre:

Grupo:

7

EXAMEN PROGRAMACION 2 - Febrero 2016

Ejercicio 1:

Escribir una clase llamada Persona:

a) Sus atributos son: nombre, edad, CI, sexo (H hombre, M mujer), peso y altura. Se pide implementar varios constructores:

- Un constructor por defecto.
- Un constructor con el nombre, edad y sexo.
- Un constructor con todos los atributos como parámetro

Implementar además getter's y setter's para sus atributos.

b) Implementar el siguiente método: esMayorDeEdad()

Devolverá un booleano según si es o no mayor de edad (mayor de edad si tiene más de 18 años)

Ejercicio 2:

a) Definir un ArrayList de Taxi (s).

```
public class Taxi {  
    private String ciudad;  
    private String matricula;  
}
```

b) Agregar 2 Taxi a el ArrayList creado en la parte anterior.

c) Buscar el/los taxi cuya Ciudad sea "Paysandu", en caso de no existir ninguno mostrar mensaje que no existe ninguno.

Ejercicio 3:

Realizar un proyecto llamado Conversor, que muestre una interfaz de usuario que nos permita realizar la conversión de Pesos a Dólares y viceversa.

La misma debe tener un botón "Convertir" y el resultado de la misma se debe mostrar en una ventana emergente.