

# 1ª GUIA DE POO

(Roberto Tecla)

Nombre: Santos Méndez Ulises Jesús

Grupo: 2CM11

Fecha: 06/05/2021

1.-Diga a que concepto corresponde la definición siguiente: denota las **características esenciales que distinguen a un objeto de otra clase de objetos** y provee así una frontera conceptual definida, relativa a la perspectiva del observador.

- a) Identidad de un objeto      b) Encapsulamiento      c) Operación      d) Abstracción      ( d )

2.- Diga a que concepto corresponde la definición siguiente: **engloba todas las** (usualmente estáticas) **propiedades del objeto** más los valores actuales (usualmente dinámicos) de cada una de sus propiedades.

- A) Estado de un objeto      B) Comportamiento de un objeto      ( A )  
C) Identidad de un objeto      D) Interfaz de un objeto

3.-Diga a que concepto corresponde la definición siguiente: Es como **actúa** y reacciona **un objeto en términos de cambio de estado**.

- A) Encapsulamiento      B) Comportamiento de un objeto      ( B )  
C) Identidad de un objeto      D) Interfaz de un objeto

4.-Diga a que concepto corresponde la definición siguiente: es la **propiedad** de un objeto que lo **distingue de todos los otros objetos**

- A) Estado de un objeto      B) Comportamiento de un objeto      ( C )  
C) Identidad de un objeto      D) Interfaz de un objeto

5.- Es una **colección de objetos que comparten estructura y comportamiento**

- a) herencia      b) clase      c) paquete      d) interfaz      ( b )

6.- Diga a que concepto corresponde la definición siguiente: es una **función** o transformación **que se puede aplicar o que puede ser aplicada** por los objetos de una clase.

- A) Identidad de un objeto      B) Encapsulamiento      ( C )  
C) Operación      D) Abstracción

7.- Diga a que concepto corresponde la definición siguiente: Es una **relación entre clases** en la que **una clase comparte la estructura y/o el comportamiento definido en una o más clases**.

- A) Herencia      B) Interfaz      ( A )  
C) Una clase abstracta      D) Encapsulamiento

8.- Es la utilización de **interfaces idénticas** con **diferentes implementaciones**.

- A) método      B) polimorfismo      C) mensaje      D) operación      E) Abstracción      ( B )

9.-De acuerdo con su **ámbito** que **variables** se pueden usar en más de un método de la misma clase

- a) Las de instancia      b) Las locales      c) Ninguna de las anteriores      ( a )

10.-De acuerdo con su **ámbito** que **variables** se declaran dentro de un método

a) Las de instancia      b) Las de clase      c) Las locales      d) Ninguna de las anteriores      ( c )

11.-Cual de las siguientes no es una **palabra reservada** en Java?

a) byte      c) catch      c) real      d) implements      e) static      ( c )

12.-Los tipos básicos en Java son:

A) String, float, double, integer, short, long, char

B) integer, byte, String, float, short, long

C) int, byte, double, float, short, long, char, boolean

D) int, short, long, char, double, float

13.-Cual de las siguientes afirmaciones no es cierta para un constructor?

A) Su tipo de retorno es void      B) Esta sobrecargado      ( A )

C) Sirve para inicializar los objetos de la clase      D) Se llama como la clase

14.-En Java \_\_\_\_\_ **solo contiene constantes y métodos abstractos**

A) Una clase abstracta      B) Una abstracción      ( A )

C) Una interfaz      D) Una operación

15.-En Java cuál de las siguientes definiciones del método suma de la clase Complejo es la correcta.

Nota: los ... representan código no mostrado.

void suma(){...}      Complejo suma(Complejo c) {...}      ( B )

Complejo suma() {...}      Complejo suma(Complejo c1, Complejo c2){...}

Diga si las afirmaciones siguientes son **verdaderas(V)** o **falsas (F)**:

1.-En POO los datos deben ir junto al código que los manipula      ( F )

2.-En POO los datos no deben ir junto al código que los manipula      ( V )

3.-La representación de un TAD solo debe ser manipulada por las operaciones definidas para ese TAD      ( V )

4.-El **encapsulamiento** sirve para ocultar la **complejidad**      ( V )

5.-El **encapsulamiento** sirve para proteger los **datos**      ( V )

6.- Si un programa en Java tiene un método **main** podemos asegurar que es una **aplicación**      ( V )

7.-Si un programa en Java tiene un método **main** podemos asegurar que es un **applet**      ( F )

8.-Si un programa en Java tiene como superclase a la clase **Applet** podemos asegurar que es un **applet**      ( V )

9.-El **estado** es lo que permite distinguir a un objeto de otro      ( F )

10.-La **herencia** en Java es múltiple      ( V )

11.-La **herencia** en Java es simple      ( V )

12.-Es la **herencia** la que permite usar métodos y atributos definidos en una superclase      ( V )

13.-La **herencia** sirve para reutilizar código      ( V )

14.-La **herencia** no sirve para reutilizar código      ( F )

15.-La **superclase** generaliza a las **subclases**      ( V )

16.-La **superclase** no generaliza a las **subclases**      ( F )

- 17.-Las **subclases** no son casos particulares de la **superclase** ( F )
- 18.-Las **subclases** son casos particulares de la **superclase** ( V )
- 19.-Si usamos **herencia** estamos haciendo uso del concepto de **abstracción** ( V )
- 20.-La POO surge para manejar el cambio y la complejidad ( V )
- 21.-La POO surge para manejar el cambio y la abstracción ( F )
- 22.-El **constructor** de los objetos de una clase es un método ( V )
- 23.-El **constructor** de los objetos de una clase no es un método ( F )
- 24.-El nombre del **constructor** de los objetos de una clase no tiene que ser el mismo que el nombre de dicha clase ( F )
- 25.-El **constructor** sirve para inicializar los objetos de una clase ( V )
- 26.-El **constructor** no sirve para inicializar los objetos de una clase ( F )
- 27.-El **constructor** de los objetos de una clase tiene como tipo de retorno **void** ( F )
- 28.-La palabra reservada **this** se usa para **resolver conflictos de nombres** ( V )
- 29.-Si 2 o más métodos tienen el mismo nombre pero distinta implementación entonces el método está **sobrecargado** ( F )
- 30.-Si hay más de un constructor en una clase entonces el **constructor** está **sobrecargado** ( V )
- 31.-Una **clase abstracta** no puede tener instancias directas ( V )
- 32.-Una **clase abstracta**, en Java, no puede tener variables de instancia ( F )
- 33.-En Java si una clase tiene un **método abstracto** entonces es una clase abstracta ( V )
- 34.-Un método **abstracto** no tiene implementación ( F )
- 35.-Un método **abstracto** no se define es decir solo se declara ( F )
- 36.-Una **interfaz**, en Java, solo contiene **constantes y métodos abstractos** ( V )
- 37.-Todo lo que está dentro de una **interfaz** es **público** ( V )
- 38.-Todo lo que está dentro de una **interfaz** es **privado** ( F )
- 39.-Una **interfaz** puede contener métodos abstractos y variables de instancia ( V )
- 40.-Una clase no puede **implementar** más de una **interfaz** ( V )
- 41.-La clase que **implementa** una interfaz no tiene que implementar todos los métodos contenidos en dicha **interfaz** ( F )
- 42.-Una **variable de clase** es compartida por todos los objetos de la clase ( V )
- 43.-En Java se pueden **concatenar cadenas** usando el operador + ( V )
- 44.-Se usa nombreObjeto.metodo(listaParametrosReales) para llamar un **método de instancia** ( V )
- 45.-Se usa nombreClase.metodo(listaParametrosReales) para llamar un **método de instancia** ( F )

### Problemas de modelado

Para cada Modelado

- Haga el diagrama de clases y en cada clase ponga los atributos y operaciones necesarias
- Muestre en dicho diagrama las relaciones entre clases (en caso de existir)

Además escribir el código en Java de las definiciones de las clases y en cada definición

- Declarar las variables de instancia (con el nivel de acceso adecuado)

- Constructor (mínimo uno)
- Codificar los métodos que sean convenientes (con el nivel de acceso adecuado)

**Problema 1.-Modelar un Carrete (o rollo) de tela**

- suponga que cada carrete viene con 200 m de tela
- la tela solo se vende al menudeo por metro (o sea no por carrete)
- Si hay suficiente tela entonces se debe cortar del carrete la cantidad de tela que pida el cliente

Carrete
+qtotal: int
+venta(qcompra:int):int

**Problema 2.-Defina el TAD para una cuenta bancaria sencilla en java.**

Aquí está la "especificación"

Cuando se abre la cuenta el cuentahabiente hace su 1er deposito (es decir deposita una cantidad de dinero inicial).

Podemos:

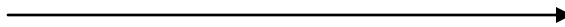
- Retirar dinero de esta cuenta.
- Depositar dinero de esta cuenta.
- Consultar el saldo (es decir el cuentahabiente puede preguntar cuánto dinero hay en su cuenta.

Cuentabnca
+nocuenta:int +saldo:double
+deposito(cdep:double):double +retirar(cret:double):double +saldof():double +imprimir:void

**Problema 3.-Codificar un applet que tenga un botón, tres campos de entrada y una etiqueta. Un campos de entrada es para las horas, otro para el numero de minutos y otro mas para el numero de segundos imprima el total de segundos transcurridos en la etiqueta si se presiona el boton. Ejemplo si se le dan 18 hrs con 3 min y 20 segundos el programa debe imprimir 65000 segundos ( $18*3600+3*60+20$ ).**

<input type="text" value="18"/>	<input type="text" value="3"/>	<input type="text" value="20"/>	<input type="text" value="65000"/>	<input type="button" value="calcula"/>
---------------------------------	--------------------------------	---------------------------------	------------------------------------	--

Ctsegundos
+calc:Button +c1:TextField +c2:TextField +c3:TextField +t1:Label +res:Label
+ init():void +actionPerformed(e: ActionEvent):void +main(args:string):static void

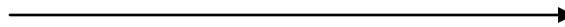


**Applet**

**Problema 4.-**Codificar un applet (o aplicación) que tenga un campo de entrada y un botón. En el campo de entrada el usuario escribe una cantidad de minutos y cuando presione el botón deben mostrarse el número de días (que hay en esos minutos) en una etiqueta y en otra etiqueta el número de horas (que hay en esos minutos).

14400	10 dias	240 horas	calcula
-------	---------	-----------	---------

Qndyh
+calc:Button +c1:TextField +t1:Label +res1:Label +res2:Label
+init():void +actionPerformed(e:ActionEvent):void +main(args:string):static void

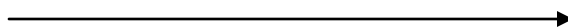


**Applet**

**Problema 5.-** Codificar un applet que tenga dos botones y dos campos de entrada como se muestra en la figura. Si el usuario presiona el primer botón la cantidad en el primer campo de entradas se convertirá a dolares y si se presiona el segundo dicha cantidad se convierte a pesos. El segundo campo se usa para meter el factor de conversión.

0	Pesos a Dolares	Dolares a Pesos	15
---	-----------------	-----------------	----

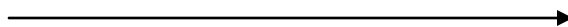
Cmoneda
+q1:JButton +q2:JButton +c1:JTextField +c2:JTextField +t1:JLabel +qdinero:double
+actionPerformed(e:ActionEvent):void +main(args:string):static void



**JApplet**

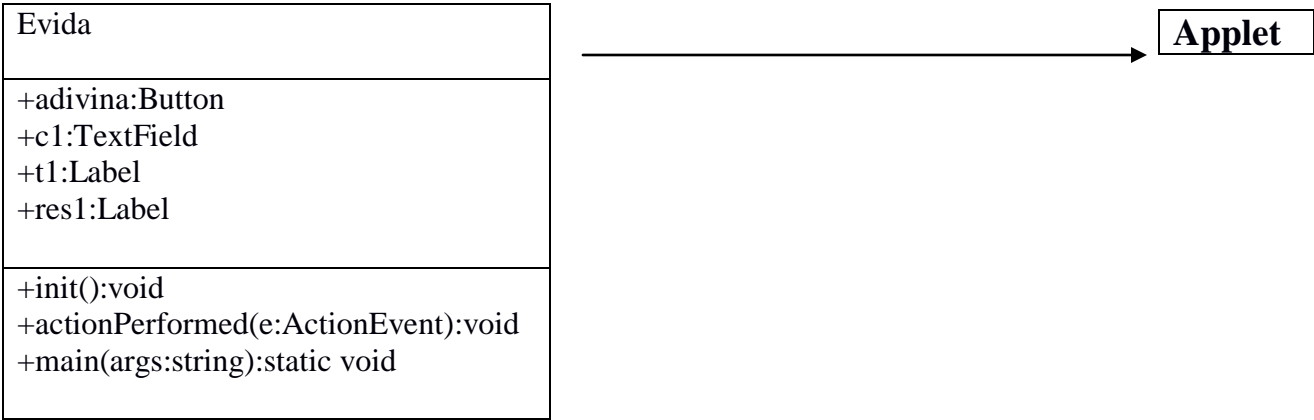
**Problema 6.-** Escriba un applet que tenga dos campos de entrada y un botón que al ser presionado muestre en una etiqueta un mensaje que diga si los contenidos de los 2 campos son iguales o no lo son.

Eiguales
+compara:Button +c1:TextField +c2:TextField +t1:Label +res1:Label
+init():void +actionPerformed(e:ActionEvent):void +main(args:string):static void

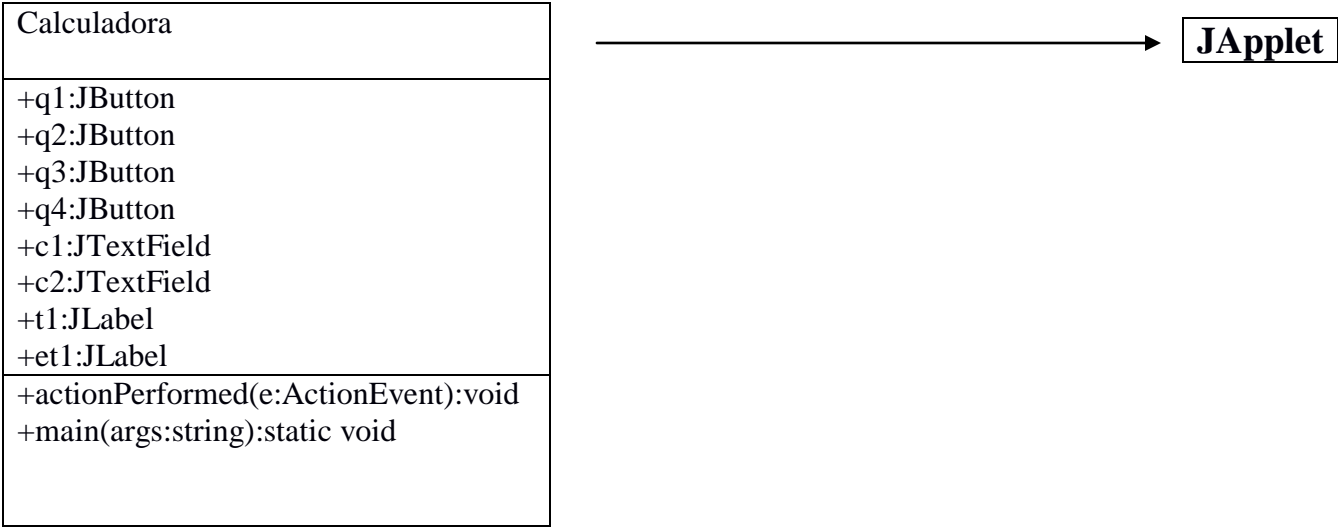
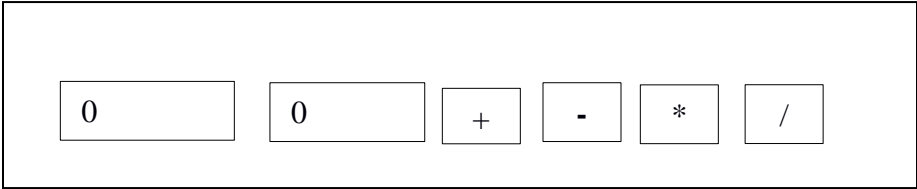


**Applet**

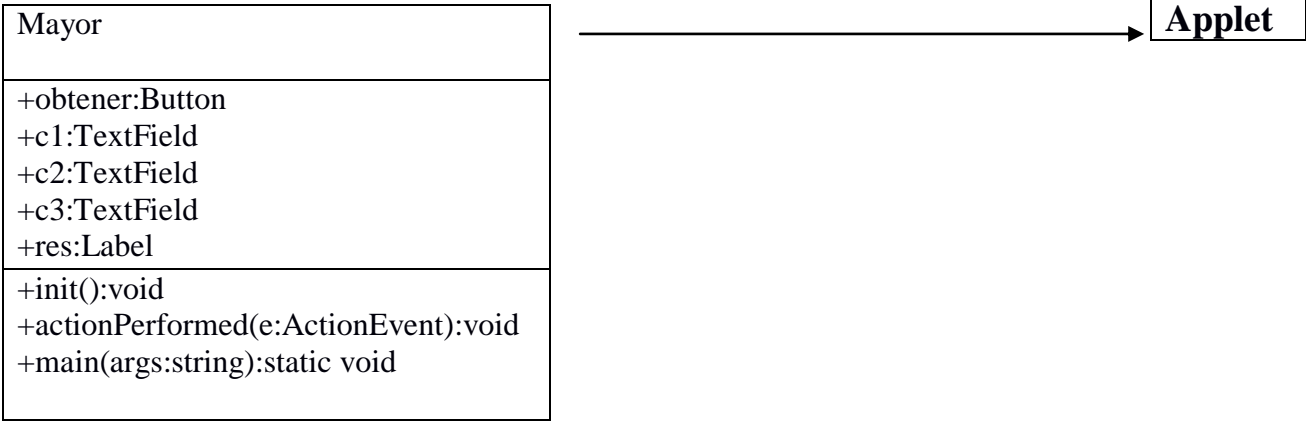
**Problema 7.-** Escriba un applet que tenga un campo de entrada donde el usuario escribirá una edad y un botón que al ser presionado muestre en una etiqueta el año en que morirá dicha persona. Suponga que la esperanza de vida es de 77 años.



**Problema 8.-** Codificar un applet que tenga dos campos de entrada y cuatro botones. Los campos de entrada se usaran para escribir dos números, el primer botón, para sumar, el segundo para restar, el tercero para multiplicar y el cuarto para dividir (ver figura). El resultado se imprimirá en la zona de status o una etiqueta (no mostrada aquí).



**Problema 9.**-Codificar un applet que tenga tres campos de entrada y un botón. Los campos de entrada se usaran para escribir tres números y cuando se presione el botón se mostrara el mayor de los 3 números en una etiqueta.



**Problema 10.**-Codificar un applet que tenga tres botones y una etiqueta, y que si se presiona un botón entonces se muestre el texto que está dentro del botón (Manzana, Limón o Naranja) en la etiqueta.

