

Universidad Autónoma de Sinaloa

Facultad de Informática Culiacán

Guía de actividades para examen ordinario y calendarización

Grupo	Fecha, Hora y Lugar
1	Jueves 11 de Enero de 2018, hora 11:00 am, Aula de Capacitación

Lineamientos de Entrega

Deberán seguir las siguientes reglas para el desarrollo de la actividad:

1. Desarrollará las actividades de resumen en un documento Word que incluya:
 - a. Hoja de presentación
 - b. Introducción
 - c. Desarrollo de cada cuestionamiento planteado
 - d. Bibliografía consultada
 - e. Conclusiones (fortalezas alcanzadas y debilidades encontradas)
2. Tanto el documento como los proyectos de desarrollo, deberán ser guardados en la carpeta compartida en Dropbox, la cual contenga su nombre completo y grupo (esta ya existe).

Instrucciones de Entrega

El próximo martes 10 de enero de 2017, subir a la carpeta Dropbox compartida, incluyendo las actividades citadas en el presente documento, así como las tareas faltantes que fueron desarrolladas durante todo el semestre, estructurando el contenido de la misma.

Actividad 1: Responder las siguientes cuestiones teóricas:

1. Defina el concepto de Lenguaje de programación (LP), cite al menos 5 ejemplos de LP.
2. Hablando en términos de Lenguajes de Programación (LP), ¿Cuál es la ventaja de estudiar más de un LP diferentes?
3. Explique cada una de las cualidades que debe tener un buen Lenguaje de Programación.
4. Explique qué diferencia existen entre los lenguajes que son Compilados e Interpretados, cite al menos 3 ejemplos de cada uno.
5. Defina al menos 5 de los 10 elementos sintácticos de un Lenguaje de programación. Cite ejemplos.

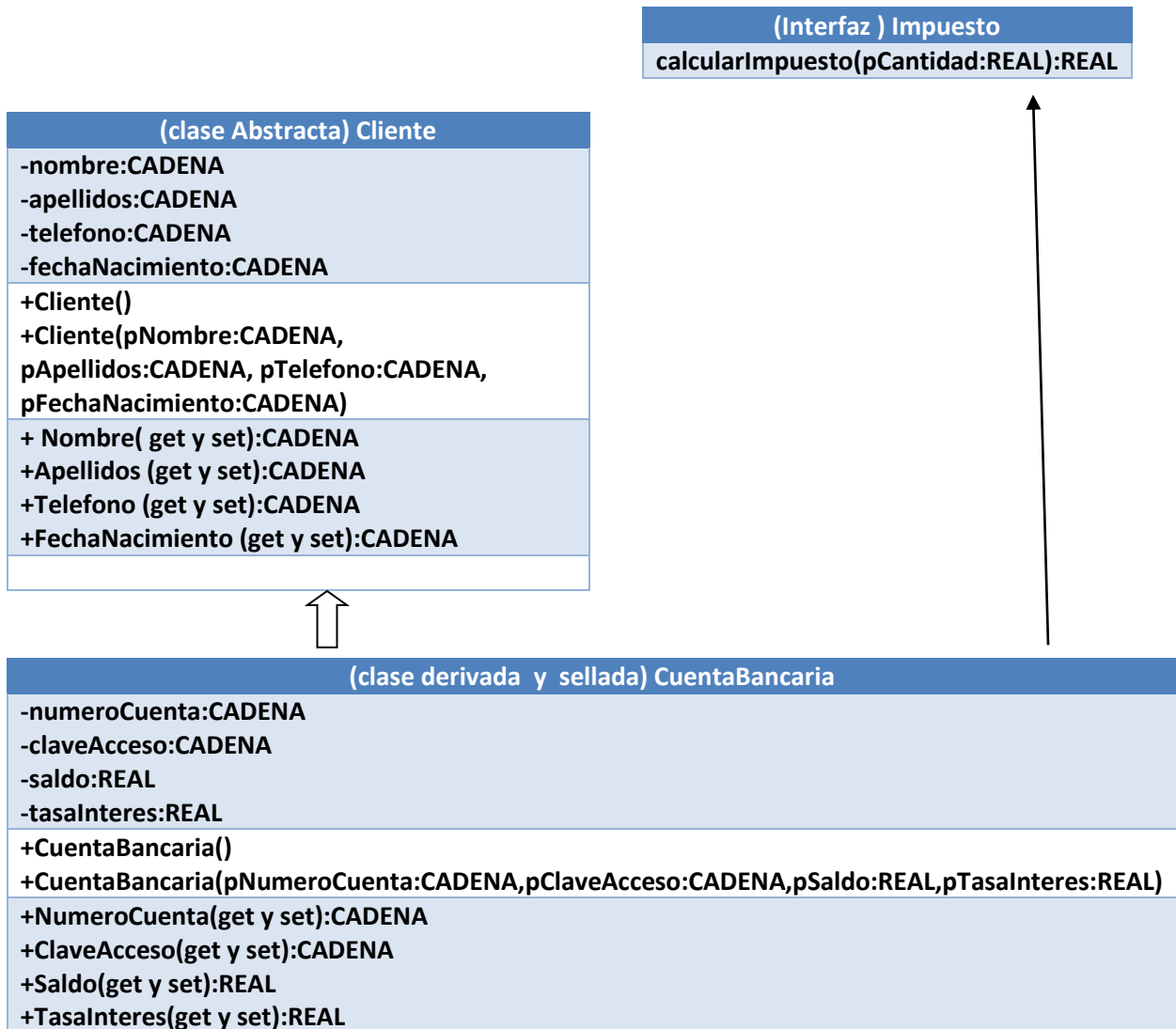
6. Describa cual es la función del analizador léxico, sintáctico y semántico, en el análisis de un programa fuente.
7. Describa los aspectos más importantes, sobre el Lenguaje de programación (LP) C# y Java.
8. Año de creación y Creador del lenguaje C# y Java.
9. Características del Lenguaje C# y Java.
10. Aplicaciones que pueden crearse o desarrollarse con los lenguajes C# y Java
11. Descripción de los componentes utilizados para la Edición, Compilación e interpretaciones según el lenguaje, de un programa fuente (Todas las etapas que atraviesa un programa fuente hasta ejecutarse en una PC).
12. Historia de Versiones del lenguaje C# y Java
13. Arquitectura de la plataforma .Net Framework para C# y la de java
14. Explicación del Modelo de Ejecución del CLR para C# y la maquina Virtual de Java
15. Herramientas utilizadas para la edición y compilación de programas fuentes tanto para C# como para Java.
16. Describa la estructura de un programa en C# y en Java
17. Como se agregan referencias de clases/bibliotecas a un programa fuente en C# y Java
18. ¿Cuáles son los Comandos utilizados para operaciones de Entrada/Salida (IO) tanto en C# como en Java?
19. ¿Cuáles son los formatos más comunes para salidas en C# y Java?
20. ¿Cómo se comenta en el lenguaje C# y Java en un programa fuente?
21. ¿Cómo se aplica (sintaxis) y cuál es la intención del tratamiento de Excepciones en C# y Java?
22. ¿Cuál es el procedimiento de compilación, ejecución y depuración por medio de comandos para un programa fuente en C# y Java?
23. Tipos de Datos más comunes del lenguajes C# y Java
24. Listado de Operadores del lenguaje C# y Java
25. Definición de constantes y variables en el lenguaje C# y Java.
26. Valores por defecto al definir variables en ambos lenguajes citados.
27. Conversión de tipos de datos en ambos lenguajes de programación
28. Lectura de datos de todos los tipos de datos en ambos lenguajes.
29. Sintaxis de las estructuras de selección y repetición en ambos lenguajes (Ejemplos de aplicación en problemas según la estructura).
30. Definición de arreglos de tamaño constante o variable según la decisión del usuario en ambos lenguajes tanto para unidimensionales como bidimensionales (ejercicios de cada caso).
31. Métodos existentes en la clase arreglos en el lenguaje C#, que nos permiten ordenar, conocer las dimensiones de las longitudes, entre otros.
32. Sintaxis de declaración de clases en ambos lenguajes.
33. Creación y destrucción de objetos en ambos lenguajes.
34. Sintaxis para constructores por defectos y parametrizados en ambos lenguajes (no olvidar el caso de aplicación en herencia).
35. Aplicación de Herencia (clase base, clase abstracta, métodos polimórficos, interfaces, clase sellada) en ambos lenguajes.
36. Aplicación de las accesibilidades private, public, protected, static en ambos lenguajes.
37. Que son las aplicaciones para Windows, cite ejemplos que se pueden desarrollar en ambos lenguajes.
38. Que son los Formularios

39. Describa los principales controles vistos para diseño y desarrollo de aplicaciones graficas tanto en C# como en Java.
40. Propiedades más importantes y uso de los siguientes controles Forms, TextBox, Label, Button, etc. vistos en el curso y sus equivalentes en Java.

Actividad 2: Codificación de algoritmos

1.- Elabore un proyecto en .NET, empleando el lenguaje c#, que dé solución al siguiente problema, empleando un proyecto gráfico (aplicación de Windows), para simular las operaciones que realizaría un cliente, en un cajero automático del **Banco FIC**. Diseñe una interfaz gráfica que simule a un cajero automático, donde el usuario (cliente) pueda realizar las diversas operaciones posibles según el siguiente diseño de clases:

Propuesta de Diseño de clases:



```
+deposito(pCantidad REAL):SINVALOR
+retiro(pCantidad REAL):BOOLEAN
+trasferencia(pCantidad:REAL):BOOLEAN
+simularInversion(años:ENTERO):REAL
```

Descripción de las operaciones:

- **+deposito (pCantidad REAL):SINVALOR.-** Consiste en aumentar el saldo, según la cantidad indicada.
- **+retiro(pCantidad REAL):BOOLEAN.-** Consiste en decrementar el saldo, según la cantidad indicada, considerando que exista suficiente saldo para realizar la operación. Se retorna un valor true, si la operación fue correcta, en caso contrario un valor false.
- **+trasferencia(pCantidad:REAL):BOOLEAN.-** Consiste en simular una transferencia de saldo, en el cual solo descontara del saldo la cantidad indicada siempre y cuando exista saldo para ello, y se cobrara un impuesto (método heredado de la interfaz (calcular impuesto)).
- **Método -calcularImpuesto(pCantidad:REAL):REAL.-** Consiste en descontar del saldo del cliente una cantidad por concepto de impuesto de la operación de transferencia, el cual se calcula en base a un porcentaje según la cantidad a transferir :
 - Un 3.5% si la cantidad <1500
 - 4% si la cantidad es de 1500 a 3000
 - 4.5 si la cantidad es mayor a 3000
- **+simularInversion(años:ENTERO):REAL.-** Consiste en calcular en base al saldo y la tasa de interés de la cuenta, la cantidad que el cliente tendría como saldo, si invierte su saldo actual sin hacer ningún otro movimiento en los años de inversión indicado, reinvertiendo los intereses al saldo cada mes. No debe modificar el saldo de la cuenta del cliente, ya que es una simulación. EL método debe retornar el total de dinero que tendría el cliente en los años de inversión indicados.

Se considerara la creatividad, innovación, originalidad del diseño de la interfaz del cajero automático, la cual interactuara con el usuario, además de la defensa de su propuesta de proyecto, la cual será parte de su evaluación ordinaria, por lo que se aconseja proteger sus proyectos y no compartirlos con sus compañeros antes de su evaluación. (La imaginación es su limitante de lo que harán, existiendo libertad para agregar más operaciones si lo considera apropiado).

Nota: Hágalo usted mismo, no busque que alguien más lo haga por usted. Suerte....

2.- Elabore un proyecto en NetBeans, empleando el lenguaje java, que dé solución al mismo problema planteado anteriormente.

Información: En java para aplicar formato de moneda, se emplea lo siguiente:

```
double importe = 1234.56;  
NumberFormat formatoMoneda = NumberFormat.getCurrencyInstance();  
System.out.println(formatoMoneda.format(importe));
```

Para mosar en un jTextField seria:

```
this(jTextField).setText(formatoMoneda.format(importe));
```