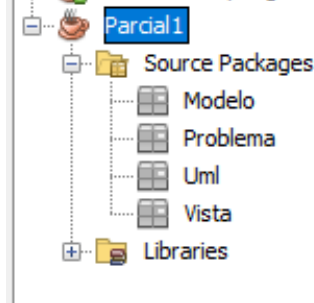


NOMBRE DE LA ASIGNATURA	Programación de Computadores II							
NOMBRE DE LA ACTIVIDAD	Parcial 1							
TIPO DE ACTIVIDAD	Sincrónica		Asincrónica	x	Individual	x	Grupal	
TEMÁTICA REQUERIDA PARA LA ACTIVIDAD			OBJETIVOS					
Unidad 1. Introducción a la POO Unidad 2. Programación basada en objetos			Desarrollo etapa de análisis, diseño e implementación de programa básico					
COMPETENCIAS			INSUMOS PARA EL DESARROLLO DE LA ACTIVIDAD / REFERENCIAS BIBLIOGRÁFICAS					
<ul style="list-style-type: none"> Identificación de clases con sus atributos requeridas en el dominio de solución de un problema Representación de clases mediante UML Identificación de relaciones de asociación (agregación y composición) entre clases Representación de relaciones entre clases mediante UML Diseño de diagramas de clases Implementación de clases mediante Java 			<ul style="list-style-type: none"> Material educativo y material complementario de la asignatura “Unidad 1.” Material educativo y material complementario de la asignatura “Unidad 2.” Talleres y código elaborado en el desarrollo de la asignatura. 					
CONOCIMIENTOS PREVIOS REQUERIDOS								
Conceptos fundamentales de POO - Estructura básica de clases – Relaciones entre clases y UML								
ESPECIFICACIONES DE LA ACTIVIDAD								
Se requiere desarrollar un programa orientado objetos en Java que de solución a la problemática planteada.								
Los principales requerimientos que debe cumplir la aplicación son los siguientes: <ul style="list-style-type: none"> El programa debe permitir la interacción mediante un Menú de opciones por consola como se muestra a continuación: 								
MENU <ol style="list-style-type: none"> Crear banco Crear sedes bancarias Consultar sedes bancaria Consultar cuenta Salir 								
<ul style="list-style-type: none"> Se debe diseñar el respectivo diagrama de clases, el cual debe incluir Clases, atributos, y relaciones entre clases (indicando nombre, navegabilidad y cardinalidad). 								

- El proyecto debe ser desarrollado en NetBeans, y debe mantener la siguiente estructura:



El código fuente debe cumplir con las siguientes prestaciones:

- Los métodos y clases deben estar debidamente comentadas.
- El código fuente debe ser legible, bien indentado y sin errores de sintaxis. Debe seguir las convenciones de Java en lo que se refiere a nombres de las clases, nombres de métodos, atributos y constantes.
- Código que no compile, no es calificado, es decir, debe ser funcional el aplicativo desarrollado.

Otras consideraciones:

- El proyecto es individual.
- El código implementado debe ser consistente con el diagrama de clases diseñado (con los atributos definidos, métodos y relaciones entre clases.)

Problema planteado:

Se desea una aplicación para la gestión de un Banco. Los Bancos ofrecen sus servicios a los usuarios desde diferentes sedes. De las sedes de un Banco se conoce el nombre, la dirección y la ciudad. En cada sede se pueden aperturar cuentas bancarias. Para aperturar una cuenta se requiere el No de cuenta, el tipo de cuenta (si es de ahorro, o corriente), el saldo inicial, y el titular de la cuenta. El titular de una cuenta bancaria es esa persona a la cual se le apertura la cuenta y los datos requeridos son su no de cc, nombre, apellido y sexo.

Por políticas del banco cada sede solo puede permitir la apertura de máximo 10 cuentas bancarias. Además, cada cuenta bancaria solo debe tener un único usuario titular.

El aplicativo debe permitir la creación de sedes y la apertura de cuentas bancarias. Así como también, mostrar la información de cualquier sede, incluyendo el listado de sus cuentas bancarias (se debe mostrar todos los datos de las cuentas bancarias de la sede, incluyendo el titular).

- **Modo de entrega**

Se debe montar en aula web enlace a repositorio GitHub del código elaborado.

**RECOMENDACIONES /
OBSERVACIONES**

Sin observaciones