

Práctica 2. Programación Orientada a Objetos Kotlin

Resolver los siguientes ejercicios utilizando el lenguaje de programación Kotlin. Realice un script por cada problema, agregar un comentario al inicio de cada script que contenga lo siguiente: Descripción corta del problema, autor, fecha creación, fecha última modificación.

Nota: Hacer uso de clases para resolver todos los ejercicios

1. **Clase para Cuenta Bancaria:** Diseña una clase CuentaBancaria que tenga un saldo y un límite de retiro. Implementa métodos set y get para establecer y obtener el saldo, y añade un método para realizar retiros que tenga en cuenta el límite de retiro. Utilice el set para validar datos.
2. **Clase para Producto:** Diseña una clase Producto que tenga un precio y un descuento aplicable. Implementa métodos set y get para establecer y obtener el precio y el descuento, y añade un método para calcular el precio final después de aplicar el descuento. Utilice el set para validar datos.
3. **Figuras:** Cree una clase abstracta "shape" la cual contenga las propiedades área, perímetro y las funciones para calcular estos valores e imprimir el resultado de cada operación. Adicional crea tres subclases cuadrado, círculo y rectángulo que reciban los valores de sus lados, el radio en el caso del círculo con constructores secundarios y heredando la clase "shape" y sus métodos. Crear instancias de las subclases y ejecutar las operaciones de área y perímetro para cada instancia.
4. **Sistema de Gestión de Biblioteca:** Diseña un sistema de gestión de biblioteca que incluya las siguientes clases e interfaces:
 - a. Material: Una clase base abstracta que tiene propiedades como título, autor, añoPublicacion, y un método abstracto para mostrarDetalles().
 - b. Libro: Una subclase de Material que tiene propiedades adicionales como género, número páginas y un método para mostrar los detalles del libro.
 - c. Revista: Una subclase de Material que tiene propiedades como issn, volumen, número, editorial y un método para mostrar los detalles de la revista.
 - d. Usuario: Una data class que tiene propiedades como nombre, apellido, edad.
 - e. IBiblioteca: Una interfaz que contenga los métodos registrar material, registrar usuario, préstamo, devolución, mostrar materiales disponibles y mostrar materiales reservados por usuario.
 - f. Biblioteca: Una clase que gestiona las operaciones de préstamo y devolución. Debe implementar la interfaz anterior. Debe tener una lista de materiales disponibles, así como un Map de usuarios donde la clave es el usuario y el valor una lista de materiales que tiene en préstamo.

- g. Crear instancias de las subclases y ejecutar las operaciones de préstamo y devolución para cada instancia.