

Programación Orientada a Objetos

Práctica N° 3 – 2025

Para cada una de los siguientes problemas plantee una solución identificando los **objetos que intervienen**, su **estado**, su **comportamiento** y las **relaciones** entre ellos.

1 - El Jardín privado

Se desea implementar un sistema de gestión de plantas para un Jardín privado, de cada planta se guarda su nombre científico, su nombre común, país de origen, fecha de compra y un identificador único. Cada vez que se crea una planta nueva para la colección, se debe asignar un número nuevo de forma automática e incremental.

2 - El convertidor de medidas

Estamos desarrollando una aplicación que realiza conversiones de unidades de medida. El sistema debe permitir convertir un valor de centímetro a pulgada y viceversa, de libra a kilogramo y viceversa, de Galón a litro y viceversa.

Tener en cuenta que existe una escala de conversión por la cual una pulgada es igual a 2,54 centímetros, una libra es igual a 0,453592 kilos y por último un galón es igual a 3,78541 litros. Definir las constantes necesarias para poder realizar las conversiones y la clase que se encargue de las conversiones no debe tener que instanciarse para poder usarse.

3 - Sistema de siembra

Una cooperativa de agricultores requiere un sistema que le aconseje cuáles son los cereales que puede sembrar en un determinado lote y qué lotes pueden ser utilizados para sembrar un determinado cereal. Para poder sembrar un cereal en un lote, éste debe contener determinados minerales. Los cereales se clasifican en granos de cosecha gruesa (como girasol, maíz,...), granos de cosecha fina (como trigo, avena,...) y pasturas (como alfalfa, trébol subterráneo,...). Un lote satisface los requerimientos de un cereal si contiene todos los minerales que requiere ese cereal. Para el caso de las pasturas, poseen la restricción **adicional** de no poder sembrarse en lotes menores a una superficie de 50 hectáreas.

Adicionalmente, los lotes son clasificados como “especiales” cuando contienen ciertos minerales de interés primario para la cooperativa, y “comunes” cuando contienen sólo minerales secundarios en la composición de la tierra. La cooperativa debe poder consultar si un lote es común o especial.

4 - Congreso de Ciencias de la Computación

El comité organizador de un Congreso de Ciencias de la Computación necesita un sistema que le ayude a asignar evaluadores de los trabajos enviados al congreso. Para esto cuenta con un conjunto de evaluadores, calificados como “generales” cuando conocen temas generales como algoritmos, lenguajes de programación, etc., o como evaluadores “expertos” cuando conocen temas específicos como agentes, visualización, redes de comunicación, etc. El sistema debe designar a un evaluador para cada uno de los distintos trabajos enviados al congreso, existiendo diferentes categorías de trabajos: artículo, resumen y poster. Cada trabajo posee un conjunto de palabras claves que definen los temas tratados en el trabajo.

Para asignar un evaluador a un trabajo, el sistema debe asegurarse que el evaluador

Programación Orientada a Objetos

Práctica N° 3 – 2025

es apto para evaluar el trabajo. Para determinar si un evaluador es apto para evaluar un trabajo, éste debe haber declarado conocimiento en todos los temas del trabajo, especificados en sus palabras claves. Para el caso de los posters, basta que el evaluador tenga conocimiento en uno de los temas del poster o sea en una de las palabras clave.

Implementar una solución orientada a objetos para este problema que permita determinar 1) qué evaluador puede asignarse a un determinado trabajo, 2) qué trabajos pueden asignarse un evaluador particular, y 3) cuántos trabajos tiene un evaluador determinado y 4) determinar si un evaluador dado es general o experto.

5 - Pila de elementos

Implementar en Java una pila de elementos. A una pila se le pueden agregar elementos utilizando el método `push(Object o)`. Para retirar elementos de la pila se utiliza el método `pop()`, que retorna el último elemento agregado y lo elimina de la misma. Es posible consultar el tope de la pila sin eliminarlo utilizando el método `top()`. La mencionada anteriormente es la única forma de consultar y retirar elementos de la pila, es decir, no se pueden obtener ni consultar elementos de otra posición que no sea el tope de la pila. Definir también los siguientes métodos:

size(): retorna la cantidad de elementos almacenados

copy(): retornar una nueva pila con una copia de los elementos de la pila original, en el mismo orden. La pila original debe mantener el orden de los elementos.

reverse(): retorna una copia de la pila pero con los elementos en el orden inverso.

6 - Modernizando el sombrero seleccionador

La Escuela Hogwarts de Magia y Hechicería decidió que era momento de reemplazar el sombrero seleccionador en la tradicional ceremonia de comienzo de año y ahora requiere de un sistema que permita realizar la asignación de alumnos a las diferentes casas. Cada una de las casas cuenta con un conjunto de cualidades que deben tener los alumnos para poder pertenecer a dicha casa y una cantidad máxima de alumnos que pueden aceptar. De esta forma, si la casa se encuentra llena, ya no puede recibir nuevos alumnos. De un alumno se conoce su nombre, su conjunto de cualidades y el conjunto de familiares que asisten a la escuela. Para que una casa pueda aceptar a un nuevo alumno, además de haber lugar, el alumno tiene que poseer todas las cualidades que la casa requiera. Es importante destacar que los alumnos pueden ser asignados como máximo a una única casa.

Algunas casas no se contentan con que los alumnos tengan todas las cualidades que requieren, sino que además en un afán de conservar los linajes de sangre pura, también exigen que el alumno cuente con un familiar que pertenezca en este momento a la misma casa. Y por último, hay casas que a lo largo de su historia se han encontrado enemistadas. En el caso en el que la casa tenga una enemistad con otra, además de exigir al alumno tener todas las cualidades, se debe controlar que dicho alumno no pueda ser aceptado por la casa enemiga.

Defina las clases para implementar una solución orientada a objetos para el problema e implemente en Java todos los métodos involucrados en asignar un alumno a una casa.

Programación Orientada a Objetos

Práctica N° 3 – 2025

7 - Mazo de cartas

Implementar un juego de cartas en el cual se dispone de un mazo y se reparten, en cada ronda, dos cartas por jugador. Un jugador gana la mano cuando los puntos de su carta son mayor a los de su oponente, en caso de empate se continúa y se realiza la suma de las cartas. El jugador con más puntos se lleva las cartas. El jugador que gana coloca todas las cartas (las suyas y las del oponente) en su pila de cartas ganadas. En caso de empate, se reparte una carta más a cada jugador hasta que uno de los dos gane. Implementar los mazos como pilas de cartas. La carta tiene un número y un palo de la baraja.