

Programación 2

Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas

Práctica N° 5 – 2020

1 - Pila de elementos

Implementar en Java una colección para almacenar una *pila* de elementos. A esta colección se le pueden agregar elementos utilizando el método `push(Object o)`. Para retirar elementos de la colección se utiliza el método `pop()`, que retorna el último elemento agregado a la colección 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 colección, 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.

2- Modernizando el sombrero seleccionador

La Escuela Hogwarts de Magia y Hechicería decidió que era momento de remplazar 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 2

Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas Práctica N° 5 – 2020

3 - Cooperativa de Agricultores

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.

Implementar una solución orientada a objetos para este problema que permita determinar:

- 1) qué cereales pueden sembrarse en un determinado lote
- 2) en qué lotes se puede sembrar un determinado cereal
- 3) determinar si un lote dado es “especial” o “común”.

4 - Sistema de Control de Gasto Público con Contribuyentes

Las ciudades citadas en el ejercicio 2 del TP1 han decidido modificar la forma en la cual obtienen el total de impuestos recaudados. Para ello cada ciudad registra una lista de los contribuyentes que aportan dichos impuestos. Los contribuyentes poseen un nombre y un código de identificación tributaria. *Hay varios tipos de contribuyentes, por ejemplo:*

- *Contribuyente simple*: abona un monto fijo en concepto de impuesto. Dicho monto fijo puede variar para cada contribuyente.
- *Contribuyente comerciante*: abona el 50% del monto fijo y el 3.5% del monto facturado en concepto de ventas.
- *Contribuyente programador*: abona el 20% del monto fijo y el 2% sobre el valor facturado por desarrollo de software.

NOTA: Incorporar las clases nuevas y realizar las modificaciones sobre las existentes para cumplir con los nuevos requerimientos. Implementar en Java.

Programación 2
Tecnicatura Universitaria en Desarrollo de Aplicaciones Informáticas
Práctica N° 5 – 2020

5 - Enunciado empleados

Una empresa solicitó crear un sistema para organizar la facturación de sus empleados. Por cada empleado la empresa almacena el nombre, el apellido, el DNI, y el sueldo mensual. Por ejemplo, el empleado Juan Hernández (DNI: 27.832.123) tiene un sueldo mensual de \$17.000. A su vez, la empresa indicó que algunos de sus empleados se dedican a vender, y para motivarlos, aparte del sueldo básico les ofrecen un extra por ventas. Este extra consiste en un porcentaje del total de ventas que haya logrado el empleado durante el mes. Por ejemplo, el empleado Martín tiene un sueldo de \$15.000, y aparte tiene un extra equivalente al 5% de todas sus ventas, en cambio, Pablo tiene un sueldo de \$13.000, pero consigue un extra equivalente al 10% de sus ventas.

- Diseñar en POO las clases necesarias para poder modelar cualquiera de los empleados. Recuerde utilizar los conceptos vistos en clase.
- A partir del listado de empleados la empresa desea poder saber la suma total de gastos en sueldos.
- ¿Cómo debería modificar mi solución para agregar nuevos tipos de empleados?

Empleado con bonos. Algunos empleados reciben un bono adicional a su sueldo si logran concretar cierto número de ventas. Por ejemplo, Julián recibe un bono de \$1000 si logra concretar al menos de 50 ventas, mientras que Mauro recibe un bono de \$3000 si concreta al menos 100 ventas.

6 - 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 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

- qué evaluador puede asignarse a un determinado trabajo,
- qué trabajos pueden asignarse un evaluador particular, y
- cuántos trabajos tiene un evaluador determinado y
- determinar si un evaluador dado es general o experto.