

TRABAJO PRÁCTICO N° 5

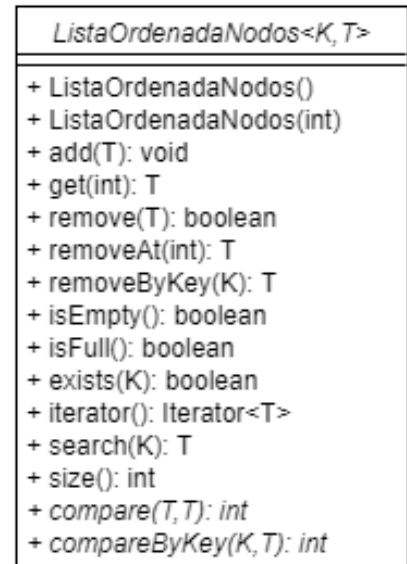
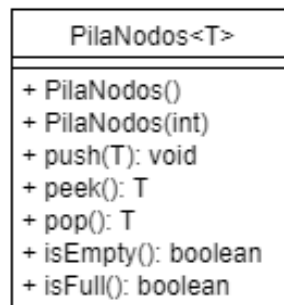
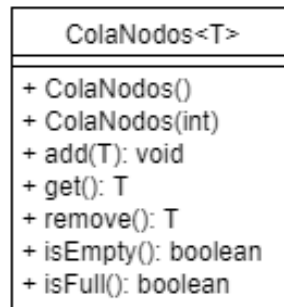
Resolvé los ejercicios utilizando diagramas de clases UML y Nassi-Shneiderman. Asegurate de leer al menos dos veces los enunciados antes de intentar confeccionar las soluciones.

Además de las estructuras de datos ya vistas (arrays y `ArrayList`), puede llegar a ser necesario utilizar las TADs `Pila`, `Cola` y `ListaOrdenada`.

Para ello, contamos con las mismas ya implementadas.

En la ilustración de la derecha se observan las correspondientes clases y sus operaciones públicas (se muestran las implementaciones que usarán en la materia **Taller De Programación I**).

En tus resoluciones, solo deberás modelar las clases que utilices con los métodos que resultaron necesarios, no hace falta incluir las implementaciones, sino las derivadas de éstas para resolver lo que se pide ([ver ejercicio 1 resuelto](#))



ENUNCIADOS

1) A cierto empleado se le encomiendan dos tareas a realizar para dos pilas de elementos que le llegan a sus manos: una de libros y otra de DVDs. De cada libro se conoce ISBN y número de páginas. De cada DVD se sabe marca y si está o no rayado.

Basado en el enunciado descripto, realizá:

El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.

La explotación del método `cantDVDsRayados` que reciba por parámetro una pila de DVDs y devuelva la cantidad de DVDs rayados que haya.

La explotación del método `paginasPromedio` que reciba por parámetro una pila de libros y devuelva el promedio de cantidad de hojas entre todos ellos.

2) En un prestigioso club de tenis se utilizan los tubos de pelotas de tenis de la firma **SmartBol** que traen de fábrica tres de ellas, las cuales son inteligentes y pueden contabilizar la cantidad de usos que tuvieron.

El criterio del club a la hora de catalogar pelotas es el siguiente:

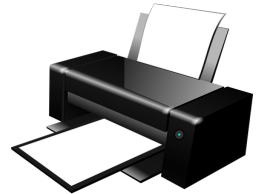
- Las pelotas sin uso obviamente son nuevas.
- Las que tienen hasta siete usos se consideran usadas
- Las que tienen más que ese valor se consideran ya gastadas.



Un tubo de pelotas se considerará usado cuando no esté lleno o al menos una de sus pelotas no sea nueva.

Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
 - B) La explotación del método **esTuboUsado** que reciba un tubo de pelotas por parámetro y devuelva si tal tubo es usado o no.
- 3) A la hora de mandar a imprimir documentos a una impresora, ésta debe imprimirlos en el mismo orden en el que fueron enviados. La impresora que vas a modelar será un poco más 'inteligente' que las demás, ya que primero verificará que haya suficientes hojas en la bandeja para imprimir todos los documentos, no quedando a mitad de camino como hacen la mayoría.



De cada documento se conoce su fecha (**String**), su cantidad de carillas y si se quiere imprimir o no en doble faz (ambas caras de la hoja). Por otro lado, la impresora dispone de los documentos a imprimir cuando se le diga, si está encendida o no y la cantidad de hojas que posee actualmente en la bandeja.

Basado en el enunciado descripto, realizá:

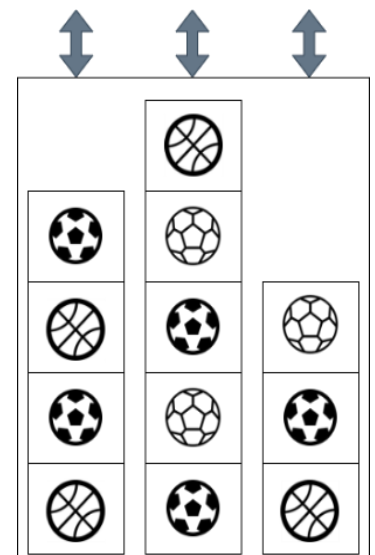
- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del método **agregarDocumento**, que permita agregar un documento a la impresora para que ésta lo imprima cuando se le diga.
- C) La explotación del método **imprimir**. Debe efectuar la impresión de todos los documentos (emularlo con una salida por consola) siempre y cuando:
 - I. Esté encendida.
 - II. Haya suficiente cantidad de hojas de papel para imprimir **todos** los documentos. Si previamente se detectara que no alcanzan las hojas, no debe imprimir ninguno.

Devuelve si se pudo imprimir o no.

- 4) El departamento de deportes de cierto colegio se encarga de prestar pelotas para las diferentes actividades físicas que allí se realizan. Las pelotas están guardadas en tres pilas, sin un orden específico. En cada pila no caben más de 5 pelotas, de las cuales se conoce para cada una su código y su tipo (fútbol, básquet o vóley).

Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del método **verPelotasEnTopes**, que muestre la información completa de las tres pelotas actualmente visibles.
- C) La explotación del método **agregarPelota**, que recibe una pelota y la guarda en la primera pila que tenga lugar para alojarla. Retorna si se pudo agregar o no.
- D) La explotación del método **buscarPelota**, que recibe un código y devuelve la pelota correspondiente (o **null** si no existe).
- E) La explotación del método **pelotasPorTipo** que retorne cuántas pelotas de cada tipo hay en cada pila.



5) Un prototipo de aplicación que gestione tickets de soporte técnico, llamado **Support**, necesita terminar de implementarse. Por ahora solo cuenta con los tickets a procesar por orden de llegada, de los cuales se sabe su ID, fecha (String), el sector de dónde proviene el reclamo (RRHH, Instituto, Escuela o Dirección) y la prioridad (alta, media o baja).

Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del método **eliminarTicket**, que reciba por parámetro un ID y borre el ticket para que no sea procesado. Debe devolver el ticket borrado o **null** si no existe.
- C) La explotación del método **cantTicketsXSectorPrioridad**, que retorne la cantidad de tickets que hay por resolver (aún sin hacerlo) por cada sector y discriminado por prioridad.

6) Cierta edificio inteligente se encarga de administrar las facturas que llegan por correo que corresponden a los servicios de los departamentos. Los atributos de cada factura son el código de barras, el número de mes, el tipo (luz, gas, Internet o ABL) y el monto a abonar.

Las facturas siempre se reciben apiladas y mezcladas, siendo el objetivo del edificio poder guardarlas en diferentes colas de forma discriminada según los tipos de servicio posibles.



Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del método **catalogar**, que reciba por parámetro la pila de facturas mezcladas y la procese (dejándola vacía) para lograr guardar las facturas encoladas según el tipo con el cuidado de que queden ordenadas por número de mes ascendente.
- C) La explotación del método **montoAAbonarPorTipo**, que retorne el monto a abonar actualmente por cada tipo de servicio.

7) Una academia ofrece una serie de cursos de diversas disciplinas, los cuales tiene guardados de una forma tal que sean accesibles muy rápidamente conociendo su código alfanumérico. Además de tal código, de cada curso se sabe su título, la cantidad de horas de duración, el nivel (básico, intermedio o avanzado) y los alumnos que posee inscriptos. De cada alumno se sabe su DNI, nombre y edad.

Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del método **obtenerEstadistica**, que reciba por parámetro el código de un curso y se retornen los siguientes datos:
 - I. Nombre del curso.
 - II. Promedio de edad de los alumnos que posee.
- C) La explotación del método **listadoDeCursosPorNivel**, que retorne una lista de cursos ordenados de menor a mayor nivel.
- D) La explotación del método **top5AlumnosMasLongevos**, de la clase **Curso** que retorne a los 5 alumnos más longevos (mayor edad) que posea (si no llega a 5, devolver los que haya). **Asumir que no hay edades repetidas.**

8) Como todos los años, es posible inscribirse voluntariamente en un centro de vacunación para recibir la dosis de la vacuna contra la gripe.

Todo aquel que sea vacunable debe ser capaz de devolver un valor entero que indica la prioridad para ser atendido. Tales valores representan: 0 para Prioridad máxima, 1 para Prioridad media y 2 para Prioridad mínima.

Serán vacunables todas las personas (de las que se pretende saber DNI, nombre, edad y si es o no paciente de riesgo). La máxima prioridad es para aquellas que sean de riesgo o cuya edad supere los 60 años. Las personas menores de 30 años tendrán prioridad mínima. El resto, prioridad media.



Hay personas consideradas esenciales, como las médicas, de las cuales interesa saber su matrícula y además siempre tendrán máxima prioridad. También interesa discriminar a personas que se desempeñen en la educación obligatoria, interesando saber en qué nivel se desempeñan (Inicial, Primario o Secundario). Este tipo de personas tendrán prioridad media salvo en el caso de quienes trabajen en nivel inicial, que tendrán prioridad máxima.

Cada centro de vacunación consta de 3 colas de candidatos a ser vacunables, los cuales se fueron inscribiendo a través de distintos medios. La idea es reordenarlas en una única cola por prioridad y, ante la misma, por orden de llegada.

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del método `obtenerColaPorPrioridadYOrden` el cual debe devolver una nueva y única cola con todos los vacunables a la espera, pero ordenados por prioridad, dejando las colas anteriores vacías. Ante igualdad en la prioridad, se desempata por orden de llegada en la cola correspondiente. Si persiste el empate, es indistinto quien va primero.

9) Es común que las aplicaciones desarrolladas en estos tiempos sean vendidas en cualquier parte del mundo, por consiguiente, en lugar de hardcodear los textos que aparecen en la aplicación (palabras, frases, etc), se obtiene cada uno a través de un diccionario de términos.

Se requiere un programa muy sencillo que le dé la bienvenida al usuario, le pregunte su nombre, lo salude utilizando tal nombre y luego se despidan. Para lograr que este programa sea multi idioma, se requiere que en el diccionario haya una lista de idiomas, y en cada idioma todos los términos que aparecen en la aplicación. Cada término tiene una clave (unívoca en cada diccionario y la misma para cada término en cada uno de los idiomas) y un valor alfanumérico con el texto correspondiente a ese término. Para el programa pedido, el diccionario quedaría así:

Clave Idioma	Clave Término	Valor
ES	BIENVENIDA	"¡Bienvenid@!"
	PEDIR_NOMBRE	"Ingresá tu nombre"
	SALUDO	"Hola"
	DESPEDIDA	"¡Adiós!"
EN	BIENVENIDA	"Welcome!"
	PEDIR_NOMBRE	"Enter your name"
	SALUDO	"Hello"
	DESPEDIDA	"Goodbye!"

Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del constructor de la clase **Saludador** en dos versiones:
 - I. Sin parámetros: El programa se ejecutará por defecto en Español.
 - II. Recibiendo por parámetro una clave de idioma: Se ejecutará el programa en tal idioma siempre que sea válido, de lo contrario, ejecutarlo en el idioma por defecto.
- C) La explotación del método **iniciarPrograma** de la clase **Saludador**, que ejecute efectivamente el programa,

10) Para las próximas elecciones la Mesa Electoral tiene una lista con todos los electores habilitados para sufragar. Esta lista ya está completa y contiene, ordenados por número de documento, los datos de cada elector, de los que también se tiene apellido, nombre y número de mesa donde debe votar. Con estos datos la Mesa Electoral genera un Padrón con todas las mesas, ordenadas por número, donde cada mesa tiene una versión más corta de los Electores (sólo número de documento, apellido y nombre) donde deben aparecer ordenados tanto por número de documento (para buscarlos) como por nombre completo (para listarlos). Puede haber nombres repetidos pero con distinto documento.

Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del método **generarElectorSinMesa** de **ElectorConMesa** que crea y devuelve una copia reducida de sí mismo (una instancia de **Elector**).
- C) La explotación del método **generarPadron** de la clase **MesaElectoral**, que genera el **PadronElectoral** con las mesas (ordenadas por número) y la versión “reducida” de los electores (ordenada también por DNI).
- D) La explotación del método **agregarElector** de **Mesa**, que recibe un **Elector** y lo guarda en las estructuras internas de **Mesa**.
- E) La explotación del método **generarRegistroElector** de **PadronElectoral**. Este método recibirá el número de dni del elector y devolverá un **RegistroDeElector** donde figuran el dni del elector, la mesa donde deberá votar y el número de orden del elector en la misma. Deben implementarse también todos los métodos que hagan falta para completarlo.

UML de muestra de Ejercicio 1 resuelto

