

TRABAJO PRÁCTICO N° 4

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.

ENUNCIADOS

1) El sistema de un restaurant en un patio de comidas cuenta con un plantel de cinco cajeros que son los que toman los pedidos de los clientes. De estos cajeros se sabe su ID entero, su apellido y antigüedad (en meses) en la empresa.

Por otra parte, se tienen ocho robots despachadores de pedidos, de los que se sabe su código alfanumérico y nivel de batería.

Por último, de los pedidos actualmente en espera se sabe el nombre de pila del cliente, el ID del cajero que lo ingresó, el código de robot que lo despachará y el monto a abonar.

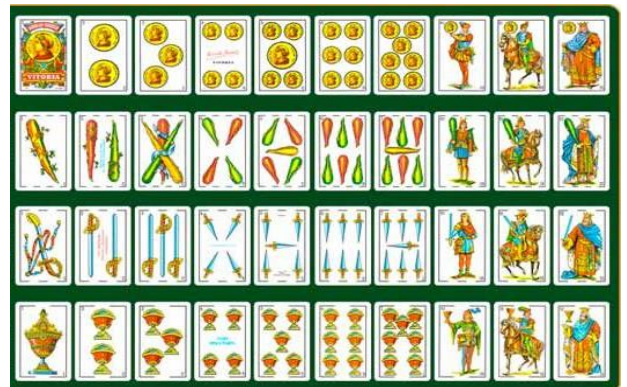
Basado en el enunciado descripto, realizá:

- A) El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- B) La explotación del método `montosPorCajero` que devuelva (no muestre por consola) el monto total de pedidos que tomó cada cajero.
- C) La explotación del método `informeDePedidos` que devuelva (no muestre por consola) un vector de objetos que contengan, por cada pedido, la siguiente información:
 - Monto del pedido.
 - Apellido del cajero que lo tomó.
 - Nivel de batería del robot asignado para despacharlo.
- D) La explotación del método `mostrarPedidosConMasRiesgo` que muestre por consola la información de los pedidos con más riesgo de no ser entregados, debido al nivel de batería de su robot despachador asignado.

2) Dada una baraja de cartas españolas sin comodines, de las que se sabe, de cada una, su número y palo, desarrollar el método `estaOrdenada` en la clase `Baraja` que permita retornar si las cartas se encuentran ordenadas tal como se ilustra en la figura.

Ayuda: Recordá que los palos se pueden enumerar. Los enumerados tienen métodos útiles que podrán ayudarte a resolver el problema.

Las cartas deben estar dispuestas de forma lineal. La imagen es ilustrativa.



3) Al mundial de fútbol la selección argentina lleva un plantel de 23 jugadores, de los cuales se conoce su posición en la cancha (arquero, defensor, mediocampista o delantero), apellido y el número de camiseta (el cual no coincide necesariamente con su ubicación en el plantel). De los 23 jugadores, es sabido que los primeros 11 lugares son para los titulares, los siguientes 7 son para los suplentes y los restantes son los de reserva.

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 **cambio** que recibe dos números de camiseta. Si no son iguales, se realiza el cambio de ubicación de los jugadores en el plantel.
 - C) La explotación del método **cambioPorLesion** que recibe el número de camiseta de un jugador lesionado y el apellido de su reemplazante. Reemplaza al lesionado por un nuevo jugador con el apellido recibido, con la misma posición y número que el que sale por lesión.
 - D) La explotación del método **obtenerReservas** que devuelva (no muestre por consola) a todos los jugadores de reserva de la selección.
 - E) La explotación del método **cantJugadoresPorPosicion** que devuelva (no muestre por consola) la cantidad de jugadores que hay por cada una de las posiciones en la cancha.
- 4) Se desea un primer prototipo de sistema de reservas de viajes en tren por el interior del país. La empresa cuenta con una lista de reservas, de las cuales se conoce su código, su recorrido y la cantidad de personas que viajarán.

Del recorrido se sabe la estación de destino y la estación de partida. Las estaciones posibles, en orden, son: **Buenos Aires, Luján, Mercedes, Suipacha, Chivilcoy, Alberti y Bragado.**

Para calcular el precio de una reserva, debe multiplicarse la cantidad de estaciones recorridas por el precio por estación, el cual es de \$50. Si el viaje fue de cabecera a cabecera (por ahora, Buenos Aires-Bragado o viceversa) se ofrece un 20% de descuento.

La empresa nos menciona que, en el futuro, está previsto que el tren pare en más pueblos y, por ende, haya nuevas estaciones, incluso intermedias, por lo que espera que el software sea mantenible.

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 **recaudacionTotal** que retorne el monto total recaudado según el precio de todas las reservas.
 - C) La explotación del método **cantVecesRecorrida** que reciba una estación por parámetro y retorne cuántas personas pasarán por esa estación, de acuerdo a las reservas existentes.
- 5) Una empresa de camiones cuenta con una flota de 10 de ellos y 20 choferes que trabajan en horarios rotativos (y rotan también por los camiones de la flota).

Por cada viaje que se realiza, se registra la siguiente información:

- ID del camión (**0-9**)
- ID del chofer (**String**)
- Kms del viaje (**double**)

Además, contamos con la siguiente información de los choferes:

- ID (**String**)
- Nombre (**String**)
- Pago por viaje (**double**)

De cada uno de los camiones, nos informan:

- ID (1 dígito, **0-9**)
- Costo por km recorrido (**double**).

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 **obtenerCostosDeViajes** que devuelva (no muestre por consola) el costo aproximado de cada viaje (honorarios + consumo).
- C) La explotación del método **obtenerViajesXChoferCamion** que devuelva (no muestre por consola) la cantidad de viajes que realizó cada chofer con cada camión.

6) Un frigorífico que fabrica embutidos, los cuales comercializa en Argentina, Chile, Uruguay y Brasil, nos pide realizar un programa para administrar sus productos y ventas. Para ello nos informa que cuenta con un plantel de 20 vendedores de los que se conoce su ID y su nombre.

El frigorífico tiene productos de los que conocemos el código, el nombre y el precio. De cada cliente sabemos su razón social, CUIT, teléfono y el país al que pertenece.

Cada vez que se toma un pedido, se ingresa en el sistema con los siguientes datos: ID (entero 0-19) del vendedor que lo realizó, CUIT del cliente que lo pidió y los productos pedidos.

Basado en el enunciado descripto, realizá:

- El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
 - La explotación del método `informarCantPedidos` que muestre por consola el nombre de cada vendedor junto a la cantidad de pedidos que ingresó por cada uno de los países.
 - La explotación del método `clientesPorPais` que retorne la cantidad de clientes por país.
 - La explotación del método `paisMasRentable` que devuelva (no muestre por consola) el nombre (`String`) del país que mayor dinero ha recaudado (se supone único).
- 7) Estás contratado a prueba por un instituto terciario para llevar a cabo tareas estadísticas y de control sobre el estado de las comisiones.

La institución guarda datos históricos sobre las comisiones donde se impartió clase. De cada una de ellas se conoce su código, la cantidad de alumnos que hubo, los datos de esos alumnos (legajo, apellido y edad) que participaron (guardados en un array) y una matriz de caracteres que representa la asistencia de cada alumno para cada clase.

Cada columna en la matriz representa una jornada de clase. Cada fila en la matriz representa a un alumno, cuyo índice coincide con el del array de alumnos paralelo, resultando en un ejemplo a nivel conceptual como el siguiente:

Apellido\Clase	1	2	3	4	5	6	7	8	9	10	11	12
Alberti	P	P	P	P	P	P	T	P	P	P	P	P
Azcúenaga	A	P	P	P	P	P	A	P	P	P	T	P
Belgrano	T	P	A	P	P	P	P	P	P	P	P	P
Castelli	P	P	T	T	A	A	A	A	A	A	A	A
Larrea	T	P	T	P	P	A	P	P	A	P	P	A
Matheu	P	P	P	P	T	T	P	P	P	P	A	P
Moreno	P	T	A	P	P	P	T	T	A	P	P	A
Paso	P	P	P	P	A	A	T	P	P	P	P	A
Saavedra	P	P	A	P	A	T	P	P	T	P	P	P

Array de alumnos

Matriz de caracteres

Cada comisión puede tener diferentes cantidades de clases y alumnos, por lo tanto, no necesariamente todas las matrices van a ser de 9x12.

Una 'P' significa Presente. Una 'T' significa Tarde. Una 'A' significa Ausente.

Un ausente suma una falta completa. Una llegada tarde suma media falta.

La condición para rendir el examen final es haber obtenido una asistencia mayor o igual al 75%.

Basado en el enunciado descripto, realizá:

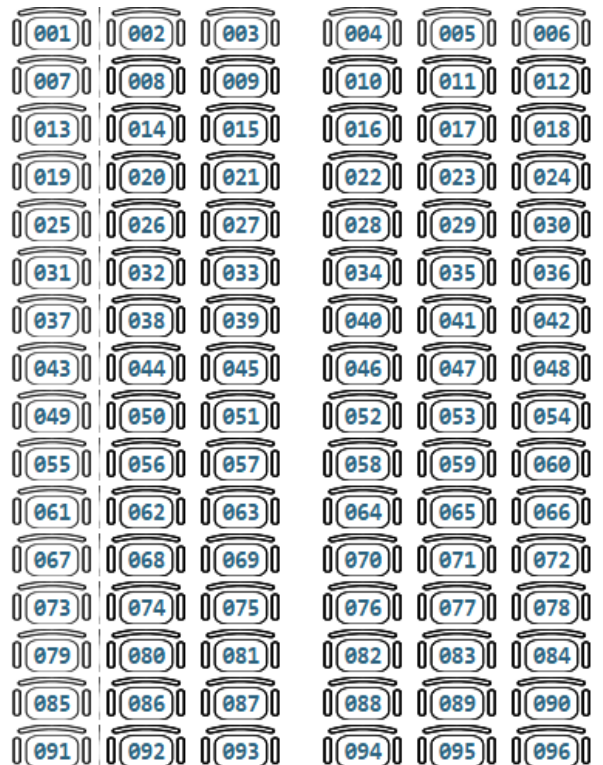
- El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- Desarrollar métodos (al menos uno por punto) en la clase `Comision` que permitan:

- I. Mostrar los apellidos de los alumnos aptos para rendir final, junto al porcentaje de presentismo de cada uno.
- II. Obtener el número total de llegadas tarde para todo el período de cursada.
- III. Obtener la cantidad de clases con asistencia perfecta de la manera más eficiente posible. (Las llegadas tarde cuentan como presente).
- IV. Devolver (**no mostrar por consola**) el porcentaje de presentismo por cada clase. (Las llegadas tarde cuentan como presente).
- V. Mostrar los apellidos de los alumnos que dejaron la cursada (más de 4 ausencias seguidas).

8) Una aerolínea nos pide diseñar un sistema para atender de la mejor manera a sus pasajeros al momento de hacer check-in. Este prototipo por ahora se centra en un único vuelo, el cual no se requiere modelar.

La aerolínea tiene una lista de reservas, ya validada, en la cual constan todas las reservas, cada una con un ID, el número de DNI del cliente que la efectuó y el número de asiento a reservar.

Por otro lado, cuenta con un solo avión del que se conoce su marca y sus asientos, los cuales están dispuestos en 16 filas de 6 asientos cada una. Cada asiento tiene su estado (libre, reservado u ocupado) y su número, el cual es determinado por la posición que ocupa en el avión, según ilustra la imagen de la derecha. Los asientos que se encuentran en las columnas de los costados de la imagen son los que están del lado de la ventanilla. Ej.: en la primera fila, los asientos **001** y **006** tienen acceso a ventanilla, los otros no.



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 **asignar** el cual cambiará de estado a reservado todos los asientos que aparezcan en la lista de reservas. La aerolínea pide que se realice de la forma más eficiente posible.
- C) La explotación del método **asientosLibresConVentanilla**, que debe devolver cuántos asientos con ventanilla están libres.

Los siguientes dos ejercicios requerirán trabajar con fechas, para eso existe una clase **Fecha** ya implementada.

Tiene la particularidad de saber reacomodar su estado si se la instancia con datos que representan una fecha incorrecta (por ejemplo, sabe transformar el **33 de enero** en **2 de febrero** y el **40 de diciembre** en **9 de enero** del año siguiente).

También puede devolver una instancia con la fecha de hoy gracias a un método de clase (**static**) llamado **hoy()**.

Esta clase está completa y funcional. Solamente debés colocarla en tu diagrama UML para relacionarla con tus otras clases del modelo y utilizarla en los diagramas Nassi-Shneiderman donde lo creas conveniente.

Fecha
+ Fecha(int,int,int)
+ hoy(): Fecha
+ getDia(): int
+ getMes(): int
+ getAnio(): int

9) Un estacionamiento de automóviles nos encomienda la creación de un prototipo de gestión para el retiro de los mismos. Dicho estacionamiento almacena los autos en espacios distribuidos de a 6 por cada uno de los 8 pisos disponibles.

De cada auto se sabe su patente, su fecha de ingreso al estacionamiento y los 12 pagos mensuales realizados en concepto de alquiler al cabo de este año, por supuesto ordenados por mes. De cada pago se conoce el monto y el DNI de la persona que lo efectuó. Para los meses aún no transcurridos o con pagos no efectuados, se guarda **null**.

A la hora de retirar un automóvil, se necesitará indicar la posición dónde se encuentra. Será recibido un solo número entero (no **String**) de dos cifras: la primera cifra indicará el piso y la segunda el espacio. Por ejemplo, el número entero **43** estaría refiriéndose al auto situado en el **tercer espacio** del **cuarto nivel**.

Cada vez que se retire un coche, se debe generar y guardar un registro donde conste la patente del mismo y la fecha de retiro. Este retiro será exitoso siempre y cuando cumpla con los siguiente:

- Que la ubicación exista.
- Que el auto no se haya ya retirado.
- Que el auto registre pagos tanto en el mes corriente como en los anteriores. Esto depende de la fecha de ingreso del mismo:
 - Si el ingreso se realizó este año, deberá controlarse que haya abonado todos los pagos desde el mes actual hasta el mes de su ingreso.
 - Si el ingreso se realizó antes del año actual, deberá controlarse que haya abonado todos los pagos desde el mes actual hasta el primer mes de este año.



El cliente nos pide expresamente que se pueda saber lo más eficientemente posible si un automóvil cumplió con todos los pagos necesarios, dándonos como dato clave que, en general, los mayores casos de morosidad se dan por no tener abonado el mes corriente.

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 **retirar**, que permita resolver lo enunciado. Retorna el auto indicado o **null**, según corresponda.
- C) La explotación del método **autosMasMorososPorPiso** que devuelva (no muestre por consola) de cada piso, el auto con mayor cantidad de pagos no efectuados durante este año (se supone único por piso).

10) El Círculo de Cine Fantástico (CCF) realiza Ciclos de Charlas con proyección de películas y debate posterior. Estos ciclos de charlas nunca tienen más de cinco charlas distintas (podrían ser menos) y siempre se realizan en días consecutivos, una por día y una por película. Una vez que estas charlas se han confirmado son inamovibles.

Debido a las problemáticas sanitarias, la cantidad de espectadores por charla no puede superar las 50 personas. Sabiendo que la cantidad histórica de espectadores suele ser cercana al doble y que se quiere hacer participar a todos, el ciclo de charlas se da en dos tandas idénticas, comenzando la segunda tanda 10 días después que la fecha de inicio del ciclo.

EJEMPLO: suponiendo que se programa un ciclo de charlas de cinco películas que comienza el 22 de septiembre, las estructuras de datos principales se distribuyen así:

Películas	"Película 1"	"Película 2"	"Película 3"	"Película 4"	"Película 5"
	↓	↓	↓	↓	↓
Tanda 1 →	Charla 1 22/9/2020	Charla 2 23/9/2020	Charla 3 24/9/2020	Charla 4 25/9/2020	Charla 5 26/9/2020
Tanda 2 →	Charla 6 2/10/2020	Charla 7 3/10/2020	Charla 8 4/10/2020	Charla 9 5/10/2020	Charla 10 6/10/2020

Para registrar un espectador a una charla, éste deberá indicar su nombre completo y el número **absoluto** de la charla que le interese (por ejemplo, si hay 2 tandas de 5 charlas cada una, empezando a contar charlas desde **1**, la segunda charla de la segunda tanda sería la número **7**).

La entrada es libre, pero debe controlarse que haya vacantes y que la persona que quiere registrarse no esté previamente registrada para ese día. Cuando se pueda registrar el asistente, recibirá un Ticket a modo de comprobante donde figuran la fecha de la charla, el título de la película y el nombre del espectador. En cualquier otro caso se retornará **null**.

Aquí se muestran algunos tickets procesados a modo de ejemplo:

```
Ticket [fecha=22/9/2020, titulo=Alien, espectador=Eloy Ente]
Ticket [fecha=4/10/2020, titulo=Encuentros Cercanos, espectador=Juan Pirulo]
Ticket [fecha=4/10/2020, titulo=Encuentros Cercanos, espectador=Eloy Ente]
Ticket [fecha=6/10/2020, titulo=El arribo, espectador=Eloy Ente]
```

Basado en el enunciado descripto, realizá:

- El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
- La explotación del constructor de la clase **CicloDeCharlas**, que debe generar las tandas y las charlas de acuerdo a los datos que llegan por parámetro:
 - El título del Ciclo de Charlas.
 - Una colección con los títulos de las películas. (Debe ser de longitud 5 o menos)
 - La fecha de inicio del ciclo. (Ver clase **Fecha** al final del enunciado)
 - La cantidad máxima de espectadores por charla. (Debe ser 50 o menos)
- La explotación del constructor de la clase **Charla**
- La explotación del método **registrarEspectadorEnCharla** de la clase **CicloDeCharlas**, cuyos parámetros, implementación y valor de retorno se explicó más arriba.
- El método **verCartelera** de la clase **CicloDeCharlas**, que permita imprimir en consola las tandas de charlas como se ilustra en el siguiente ejemplo:

```
Ciclo de Charlas 'Visitantes' - 22/9/2020
** Tanda 1 **
22/9/2020    Charla  1      Alien          (49 vacantes)
23/9/2020    Charla  2      Mars Attack    (50 vacantes)
24/9/2020    Charla  3      Encuentros Cercanos (50 vacantes)
25/9/2020    Charla  4      Avatar          (50 vacantes)
26/9/2020    Charla  5      El arribo       (50 vacantes)
** Tanda 2 **
2/10/2020    Charla  6      Alien          (50 vacantes)
3/10/2020    Charla  7      Mars Attack    (50 vacantes)
4/10/2020    Charla  8      Encuentros Cercanos (48 vacantes)
5/10/2020    Charla  9      Avatar          (50 vacantes)
6/10/2020    Charla 10      El arribo       (49 vacantes)
```

(OPCIONAL)**EJERCICIOS SIMPLES CON ARRAYS PARA ESTIMULAR LA LÓGICA**

Desarrollá la clase `UtilArrays`, sin atributos, que cuente con métodos estáticos que permitan:

- 11) Mostrar el contenido de un array de caracteres recibido como parámetro sin valores repetidos.

```
public static void mostrarSinRepetidos (char[] elementos)
```

`'I' 'N' 'M' 'I' 'N' 'E' 'N' 'T' 'E'` ➡ `'I' 'N' 'M' 'E' 'T'`

- 12) Retornar si el contenido de un array de caracteres recibido como parámetro representa un palíndromo.

```
public static boolean esPalindromo (char[] elementos)
```

`'I' 'N' 'M' 'I' 'N' 'E' 'N' 'T' 'E'` ➡ `false`

`'R' 'E' 'C' 'O' 'N' 'O' 'C' 'E' 'R'` ➡ `true`

- 13) Invertir el contenido de un array de caracteres recibido como parámetro, sin usar un array auxiliar.

```
public static void invertir (char[] elementos)
```

`'I' 'N' 'M' 'I' 'N' 'E' 'N' 'T' 'E'` ➡ `'E' 'T' 'N' 'E' 'N' 'I' 'M' 'N' 'I'`

- 14) Rotar una posición hacia la derecha cada elemento de un array de caracteres recibido como parámetro, sin usar un array auxiliar.

```
public static void rotar1 (char[] elementos)
```

`'I' 'N' 'M' 'I' 'N' 'E' 'N' 'T' 'E'` ➡ `'E' 'I' 'N' 'M' 'I' 'N' 'E' 'N' 'T'`

- 15) Mostrar los valores resultantes de la unión de dos arrays de caracteres (sin repeticiones) recibidos como parámetros.

```
public static void mostrarUnion (char[] cjto1, char[] cjto2)
```

`'B' 'L' 'A' 'N' 'C' 'O'` ➡ `'B' 'L' 'A' 'N' 'C' 'O' 'S' 'M'`

`'S' 'A' 'L' 'M' 'O' 'N'`

- 16) Mostrar los valores resultantes de la intersección de dos arrays de caracteres (sin repeticiones) recibidos como parámetros.

```
public static void mostrarInterseccion (char[] cjto1, char[] cjto2)
```

`'B' 'L' 'A' 'N' 'C' 'O'` ➡ `'L' 'A' 'N' 'O'`

`'S' 'A' 'L' 'M' 'O' 'N'`

- 17) Mostrar los valores resultantes de la diferencia de dos arrays de caracteres (sin repeticiones) recibidos como parámetros (el primero menos el segundo).

```
public static void mostrarDiferencia (char[] cjto1, char[] cjto2)
```

'B'	'L'	'A'	'N'	'C'	'O'
-----	-----	-----	-----	-----	-----

➡ 'B' 'C'

'S'	'A'	'L'	'M'	'O'	'N'
-----	-----	-----	-----	-----	-----

- 18) Mostrar los valores resultantes de la diferencia simétrica de dos arrays de caracteres (sin repeticiones) recibidos como parámetros (el primero menos el segundo).

```
public static void mostrarDiferenciaSimetrica (char[] cjto1, char[] cjto2)
```

'B'	'L'	'A'	'N'	'C'	'O'
-----	-----	-----	-----	-----	-----

➡ 'B' 'C' 'S' 'M'

'S'	'A'	'L'	'M'	'O'	'N'
-----	-----	-----	-----	-----	-----

- 19) Una pyme, para ser considerada como tal, no debe superar los 665 empleados en su nómina. De cada empleado se conoce su DNI, apellido y sueldo. Al ser muy común el alta y la baja de sus trabajadores, la pyme requiere un sistema que mantenga actualizada la nómina de forma eficiente.

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 **altaDeEmpleado** que recibe los datos de un empleado como parámetro y lo agrega a la primera posición disponible de la nómina. Se retorna el número de posición del nuevo empleado (o **-1** si la nómina estaba completa).
- C) La explotación del método **bajaDeEmpleado** que recibe el DNI de un empleado como parámetro y lo quita de la nómina (colocando un **null** en esa posición). Devuelve al empleado dado de baja o **null** si no se encontró.
- D) La explotación del método **cantVacantes** que retorne cuántas vacantes hay en la nómina.
- E) La explotación del método **sueldoTotal** que retorne el sueldo total de todos los empleados.

- 20) Como en el último tiempo la nómina de empleados viene siendo más estable, la misma pyme ahora nos pide que le demos más prioridad a la eficiencia del proceso para saber la cantidad de vacantes y el sueldo total. Nos pide que en lugar de tener que recorrer los 665 lugares chequeando valores en **null**, mejor exista una manera de ciclar tantas veces como empleados realmente haya.

Realizá los cambios que consideres necesarios en los métodos para cumplir con lo pedido.