

TRABAJO PRACTICO Nº 5

PARADIGMAS DE PROGRAMACIÓN

Dr. Pablo Javier Vidal

Unidad 2

Ejercicio Integrador

Para cada uno de los ejercicios debe desarrollar el diagrama de clases completo. Debe evaluar el funcionamiento total del programa con una clase llamada **TestRunner.java** que evalúe donde instancie los diferentes componentes de su sistema y muestre como interaccionan entre si. De ser necesario puede agregar métodos extras para mostrar información por sistema o las clases necesarias para completar el programa según su entendimiento del problema.

Ejercicio 1.

Realice un programa en Java que pruebe el funcionamiento de estas clases. Debe crear objetos y comprobar el correcto funcionamiento de todos los métodos definidos.

1. Defina una clase *Forma* que tenga los siguientes atributos:

- Color
- Coordenada del centro de la forma (Punto)
- Nombre de la forma

Y, al menos, los siguientes métodos:

- Imprimir
- Obtener y cambiar el color
- Mover la forma (o sea, su centro)

2. Defina una clase derivada *Rectangulo* que tenga los siguientes atributos:

- Lado menor.
- Lado mayor.

Y, al menos, los siguientes métodos:

- Imprimir: Debe imprimir qué se trata de un rectángulo mostrando su nombre, color, centro y tamaño de lados. Debería usarse la función Imprimir de la clase base para realizar parte de este trabajo.
- Calcular el área
- Calcular el perímetro
- Cambiar el tamaño del rectángulo. Recibe como parámetro un factor de escala. Así, por ejemplo, si el factor vale 2, el rectángulo duplicará su tamaño y si es 0,5 se reducirá a la mitad.

3. Defina una clase *Elipse* derivada de *Forma*. Recordatorio: una elipse queda definida por su radio mayor (R) y su radio menor (r), tal que el área de una elipse es igual a $\pi \cdot (R \cdot r)$.
4. Defina una clase *Cuadrado* derivada de la clase *Rectangulo*.
5. Defina una clase *Circulo* derivada de la clase *Elipse*.
6. Definir el diagrama de clases en base a lo anteriormente planteado.

Evaluar :

- ¿Cómo haría para obligar que todas las clases futuras derivadas tengan al menos los métodos *area* y *perimetro*. ¿Tiene sentido incluir dichos métodos en la clase *Forma*?
- Diseñe un diagrama de clases que refleje la estructura definida hasta el momento. Añádale las clases: *Punto*, *Línea*, *Triángulo*, *Triángulo Rectángulo* y *Polígono*. ¿dónde irían? ¿Cuáles serían sus atributos y propiedades? (No los implemente).

Ejercicio 2.

Se pretende desarrollar un conjunto de clases que representen, de forma simplificada, a una hipotética empresa dedicada a vender un producto. A continuación, se describen las características básicas de estas clases:

1. *Empleado*: Clase básica que describe a un empleado. Incluye sus datos personales (nombre, apellidos, DNI, dirección) y algunos datos tales como los años de antigüedad, teléfono de contacto y su salario. Se debe incluir también información de quién es el empleado que lo supervisa.

La clase debe tener contener al menos, las siguientes métodos:

- Constructores para definir correctamente un empleado, a partir de su nombre, apellidos, DNI, dirección, teléfono y salario.
- Imprimir (A través de los operadores de E/S (`System.out.println()`) imprimir los datos básicos)
- Cambiar supervisor
- Incrementar salario

2. *Secretario*: Tiene despacho, número de fax e incrementa su salario un 5 % anual. Tendrá, al menos, las siguientes funciones miembro:

- Constructores (debe rellenar la información personal y los datos principales)
- Imprimir (debe imprimir sus datos personales y su puesto en la empresa).

3. *Vendedor*: Tiene auto de la empresa (identificado por la matricula, marca y modelo), teléfono celular, área de venta, lista de clientes y porcentaje que se lleva de las ventas en concepto de comisiones. Incrementa su salario un 10 % anual.

Debe tener, al menos, las siguientes funciones miembro:

- Constructores (debe rellenar la información personal y los datos principales)
- Imprimir (debe imprimir sus datos personales y su puesto en la empresa).
- Dar de alta un nuevo cliente.
- Dar de baja un cliente.
- Cambiar de coche.

4. *Jefe de zona*: Tiene despacho, tiene un secretario a su cargo, una lista de vendedores a su cargo y tiene auto de la empresa (identificado por la matrícula, marca y modelo). Incrementa su salario un 20 % anual.

Debe tener, al menos, las siguientes funciones miembro:

- Constructores (debe rellenar la información personal y los datos principales)
- Imprimir (debe imprimir sus datos personales y su puesto en la empresa).
- Cambiar de secretario.

- Cambiar de coche.
- Dar de alta y de baja un nuevo vendedor en su zona.

Se solicita:

- Que todas las instancias a implementar sean empleados de tipo vendedores, jefes de zona o secretarios.
- Diseñar el diagrama de clases de acuerdo a lo anteriormente explicado.
- Hacer una clase de prueba que debe evaluar cada una de las clases implementadas.

Ejercicio 3.

Se pretende realizar una aplicación para esta facultad que gestione la información sobre las personas vinculadas con la misma, que se pueden clasificar en tres tipos: estudiantes, profesores y personal de servicio.

A continuación, se detalla qué tipo de información debe gestionar esta aplicación:

- Por cada persona, se debe conocer, al menos, su nombre y apellidos, su número de identificación y su estado civil.
- Con respecto a los empleados, sean del tipo que sean, hay que saber su año de incorporación a la facultad y qué número de legajo tienen asignado.
- En cuanto a los estudiantes, se requiere almacenar el curso en el que están matriculados.
- En lo que se refiere a los profesores, es necesario gestionar a qué departamento pertenecen (historia, matemáticas, arquitectura, ...).
- Sobre el personal de servicio, hay que conocer a qué sección están asignados (biblioteca, decanato, secretaría, ...).

El ejercicio consiste, en primer lugar, en definir la diagrama de clases y las diferentes relaciones que se encuentran para esta aplicación. A continuación, debe programar las clases definidas en las que, además de los constructores, getters y setters, hay que desarrollar los métodos correspondientes a las siguientes acciones:

- Cambio del estado civil de una persona.
- Reasignación de despacho a un empleado.
- Matriculación de un estudiante en un nuevo curso.
- Cambio de departamento de un profesor.
- Traslado de sección de un empleado del personal de servicio.
- Imprimir toda la información de cada persona

Incluya una clase de prueba que permita instanciar los diferentes tipos de clases y probar los métodos desarrollados.

Ejercicio 4.

Se plantea extender el ejercicio anterior incluyendo una clase que represente al centro docente. Esa clase incluirá 3 contenedores, uno por cada tipo de persona vinculada con el centro.

- Se deben incluir los siguiente métodos:
 - Uno para dar de alta una persona, que incorporará a la persona en la lista correspondiente.
 - Otro para dar de baja una persona, dado su DNI. Añada un método a la clase persona para poder obtener el DNI de un objeto de esa clase.
 - Uno para imprimir toda la información de las personas vinculadas con el centro.

Ejercicio 5.

Se pretende desarrollar una aplicación que permita calcular los precios de alquiler de una empresa de alquiler de vehículos.

Cada vehículo se identifica unívocamente por medio de su matrícula.

La empresa alquila distintos tipos de vehículos, tanto para transporte de personas como de carga. En la actualidad los vehículos alquilados por la empresa son: autos, buses, camionetas de carga y camiones.

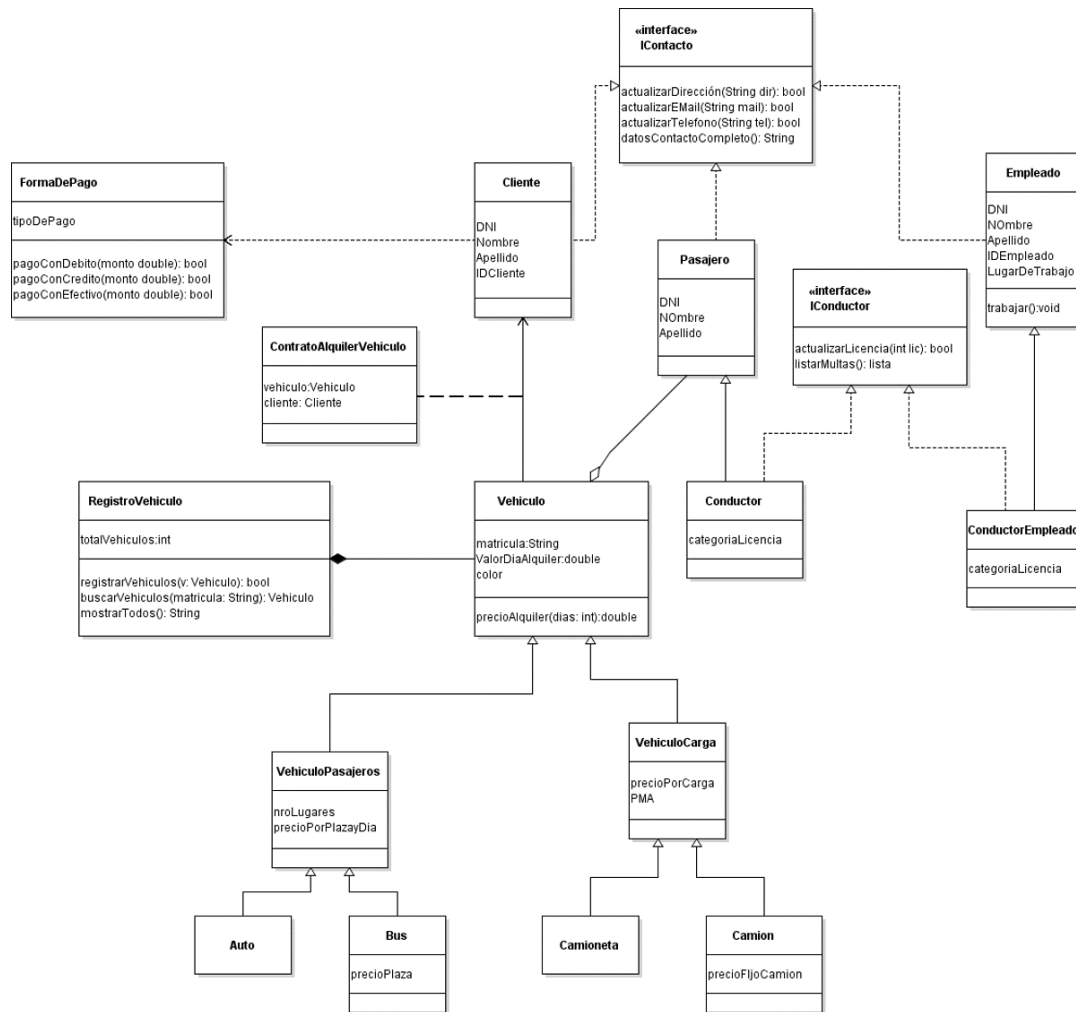
El precio del alquiler de cualquier vehículo tiene una componente base que depende de los días de alquiler a razón de 500 pesos/día. En el caso de alquiler de un auto, al precio base se le suma la cantidad de 150 pesos por butaca y día.

El precio de alquiler de los buses es igual que el de los autos, salvo que se le añade una cantidad de 350 pesos por butaca independientemente de los días de alquiler.

El precio de los vehículos de carga es el precio base más $1000 + (1000 * PMA)$ (PMA=peso máximo autorizado en toneladas). Además, en el caso de los camiones, al precio se suma un fijo de 2500 pesos independientemente de los días de alquiler. La interacción del empleado con la aplicación deberá realizarse a través de una interfaz (ya sea gráfica o de opciones básicas) basada en menú.

Un cliente puede reservar un vehículo y la forma de pago puede ser en efectivo, con tarjeta de débito o crédito. Asimismo, la empresa provee un conjunto de conductores para manejar los vehículos o bien uno de los pasajeros se designa como conductor.

Dado el siguiente diagrama de clases, implemente cada una de ellas con los atributos y métodos declarados. De ser necesario agregue los atributos y/o métodos que considere necesario.



1. Nombrar las interfaces identificadas.
2. Nombrar aquellas clases que según ud. pueden llegar a ser abstractas en el modelo presentado.
3. Identificar las relaciones existentes dentro del diagrama de clases nombrando las clases, la cardinalidad y el rol en cada relación (incluida herencia).
4. Definir los atributos que necesitan listas para trabajar. Implementar los métodos que permitan administrar esas listas
5. Definir las excepciones en aquellos casos que Ud. considere necesario. Al menos debe haber dos controles de excepciones con sus correspondientes cláusulas
6. Agregar al diagrama e implementar una clase que permita generar facturas, ya sea vía impresión de información (vía pantalla) o vía persistencia en disco (guardar un archivo binario).
7. Definir una serie de opciones que permitan al usuario poder alquilar al menos un auto y luego generar la factura correspondiente. Para ellos definir una clase con un metodo *main* que les permita crear un listado de vehiculos, uno de clientes y al menos un contrato del que puedan imprimir la factura correspondiente.

Ejercicio 6.

Una pizzería de la ciudad ofrece a sus clientes una amplia variedad de pizzas de fabricación propia, de varios tamaños (8, 10 y 12 porciones).

Los clientes tienen a disposición un menú que describe para cada una de las variedades, el nombre, los ingredientes y el precio según el tamaño y el tipo (a la piedra, a la parrilla, de molde) de la pizza. Los clientes realizan sus pedidos en el mostrador.

El pedido debe contener el nombre del Cliente, para llamarlo cuando su pedido está listo; la cantidad de pizzas, el tamaño, la variedad, la fecha del pedido, la hora en la que el pedido debe entregarse y la demora estimada informada al cliente.

El pedido va a la cocina y cuando está preparado se informa al que lo tomó para que se genere la factura correspondiente y se le entregue el pedido al cliente.

El dueño de la pizzería ha manifestado la necesidad de acceder al menos a la siguiente información:

- Variedades y tipos de pizzas más pedidas por los clientes.
- Ingresos (recaudaciones) por períodos de tiempo.
- Pedidos (cantidad y monto) por períodos de tiempo.

Ejercicio 7.



Un día normal, Red, Chuck y los Blues cuidaban su nido y sus huevos, mientras que por allí cerca pasaba una expedición conformada por el Rey Cerdo, y unos cerdos obreros buscando comida. Ven a lo lejos unos huevos y se les antojan. Entonces, Red y sus compañeros se distraen. Después de un rato se dan cuenta de que los huevos no están y ven a los cerdos cocinándolos. Así se inicia una guerra entre los Pájaros y los Cerdos.

■ Cuidado con los pájaros

Todos los pájaros de la isla quieren atacar a los cerdos y para eso necesitan tener fuerza. Además, también pueden enojarse, lo cual los hace más fuertes. Esto en general implica solo duplicar su ira, salvo algunos casos puntuales que se mencionan. Los pájaros se comportan de la siguiente manera:

- Pájaros comunes: Para todos estos pájaros, la fuerza es el doble de su ira.
- Red: Su fuerza es su ira * 10 * la cantidad de veces que se enoja hasta el momento, ya que es rencoroso.
- Bomb: Su fuerza es el doble de su ira, siempre que no supere su tope máximo de fuerza permitido, que actualmente es de 9000 pero podría cambiar.
- Chuck: Su fuerza depende de la velocidad a la que esté corriendo en ese momento. Hasta los 80 kms por hora, es de 150, luego se le suma 5 * cada kilómetro que se pase de 80. Cuando se enoja, duplica su velocidad.
- Terence: Es otro pájaro rencoroso cuya fuerza depende de su ira, de la cantidad de veces que se enoja y un multiplicador que puede cambiar.

- Matilda: Su fuerza se calcula como el doble de su ira más la suma de la fuerza de todos sus huevos (cada huevo tiene la fuerza de su propio peso). Al enojarse, pone un huevo de 2 kilos.

■ Una isla siniestra

En la isla Pájaro ocurren diferentes eventos que afectan la vida de nuestros pájaros favoritos. Entre ellos se encuentran:

- Sesión de manejo de la ira con Matilda: Tranquiliza a todos los pájaros de la isla. Esto implica disminuir su ira en 5 unidades.
- A Chuck nada lo tranquiliza.
- Invasión de cerditos: Enoja a todos los pájaros una vez por cada 100 cerditos que invaden.
- Fiesta sorpresa: Hace enojar solamente a los homenajeados. ¿Qué pasa si no hay ninguno?
- Serie de eventos desafortunados: Suceden varios eventos de los anteriores, secuencialmente.

■ Guerra porcina

Los cerditos se robaron todos los huevos de los pájaros y los tienen escondidos en la isla Cerdito. Para recuperarlos, los pájaros van a tener que pelear, y derribar todos los obstáculos que encuentren en su camino. Algunas cosas con las que pueden toparse son:

- Paredes de vidrio: Tienen una resistencia de 10 por el ancho de la pared.
- Paredes de madera: Tienen una resistencia de 25 por el ancho de la pared.
- Paredes de piedra: Tiene una resistencia de 50 por el ancho de la pared.
- Cerditos obreros: Tienen 50 de resistencia.
- Cerditos armados: Los cerditos con casco tienen tanta resistencia como $10 \times$ la resistencia de su casco. Los cerditos con escudo resisten $10 \times$ la resistencia de su escudo.
- Cuando la isla Pájaro ataca a la isla Cerdito, lanza de a uno a todos sus habitantes, quienes impactan en el obstáculo más cercano. Si la fuerza del pájaro es mayor a la resistencia del obstáculo, lo derriba. Si no, el obstáculo se mantiene en pie.

■ ¿Y si se abren los huevos?

Si quisiéramos incorporar nuevos pájaros a la isla, ¿qué cambios habría que hacer en la solución planteada? ¿Qué conceptos nos ayudan?

Aparte del diagrama de clases y de la implementación completa de cada clase se pide:

- Obtener la fuerza de un pájaro.
- Hacer enojar a un pájaro.
- Obtener a los pájaros fuertes de la isla, que son los que tienen una fuerza de más de 50.
- Obtener la fuerza de la isla, que es la suma de la fuerza de todos sus pájaros fuertes.
- Saber si un pájaro puede derribar un obstáculo.
- Hacer que la isla Pájaro ataque a la isla Cerdito.
- Saber si se recuperaron los huevos, es decir, si la isla Cerdito quedó libre de obstáculos.
- Hacer que suceda un evento cualquiera en la isla pájaro.

Marco Teórico

Ejercicio 8.

Explique qué restricciones debería tener un lenguaje orientado a objetos para que su chequeo de tipo pueda realizarse estáticamente. ¿Qué desventajas tiene aplicar tales restricciones?

Ejercicio 9.

¿Qué diferencias ve entre el sistema de tipos de Java y el C#? Realice un análisis comparativo entre estos lenguajes utilizando los criterios y/o características vistos en la materia.

Ejercicio 10.

Explique porque en los lenguajes orientados a objetos es necesario que la ligadura entre un mensaje (llamada) y el método correspondiente se resuelva dinámicamente.

Ejercicio 11.

¿Qué consideraciones deberían tomarse para modelar adecuadamente un lenguaje de programación orientado a objetos que tiene tipado dinámico como Python?