



Paradigmas y Lenguajes de Programación II

Actividad Práctica: conocimiento entre objetos

Objetivo:

- Familiarizarse con los objetos y sus relaciones.
- Comprender el proceso de instanciación de objetos y el envío de mensajes entre ellos.
- Interpretar diagramas de clase representados en UML

EJERCICIO 1: La empresa Visual-Dron que se dedica al alquiler de drones para filmaciones, desea simular los alquileres de sus Drones.

Para la simulación de alquileres debe tener en cuenta lo siguiente:

Precio por tipo de filmación:

Tipo de filmación	Costo por hora
4k	\$1500
FULL-HD	\$1200

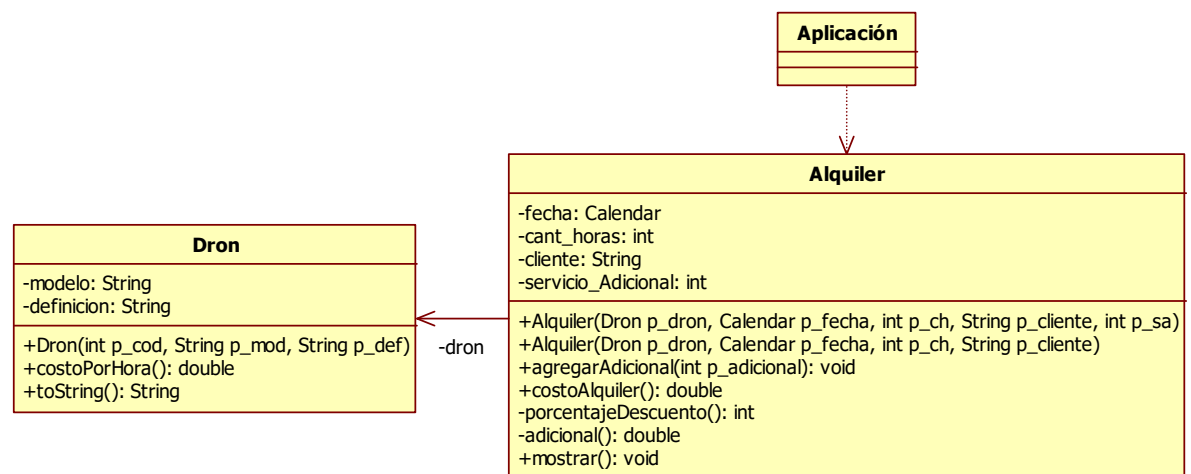
Adicionales:

N°	Servicio	Costo adicional
1	Filmación 360°	\$2000
2	Transmisión en vivo	\$1000

Descuentos por horas de filmación:

Cantidad de horas	Descuento
5 horas o más	25 %
De 3 hasta 5	15 %

Teniendo en cuenta el diagrama de clases:





- Implementar en java las clases Dron y Alquiler con sus constructores, setter, getter.
- El método toString () de la clase Dron, devuelve un String con el siguiente formato:
Modelo de Dron: DJI Phantom 3 – Definición: 4k - Costo por Hora: \$ 1500.00
- El método agregarAdicional() de la clase Alquiler, permite agregar un servicio adicional al alquiler.
- El método porcentajeDescuento() de la clase Alquiler, retorna el porcentaje que se descontará, si correspondiera, sobre el total a pagar.
- El método costoAlquiler() devolverá el costo final que se deberá abonar.
- El método mostrar () de la clase Alquiler, debe tener la siguiente salida por pantalla:

***** Empresa Visual-Dron*****

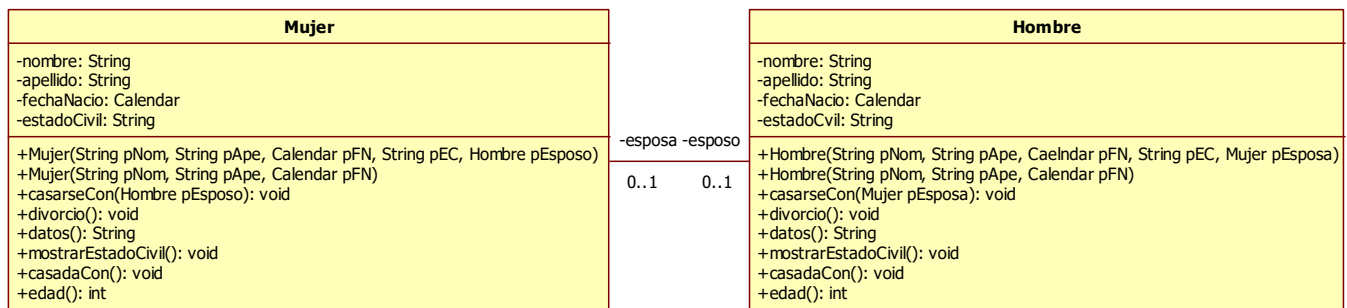
Modelo de Dron: DJI Phantom 3 – Definición: 4k - Costo por Hora: \$ 1500.00

Fecha de alquiler: 12/02/2018 – Cliente: Marcos Alarcon - Hs de alquiler: 4

Adicional por 360°: \$2000 – Descuento del 15 %- Costo final: \$ 6800

Crear una clase ejecutable que permita instanciar los objetos, ingresando los datos a por teclado, permitiendo simular el alquiler de drones y probar exhaustivamente los métodos y mostrar los resultados de la simulación por pantalla.

EJERCICIO 2: El Registro Civil de la Ciudad desea automatizar la emisión de certificados de matrimonio y divorcio, para lo cual necesita administrar la información de hombres, mujeres y la relación entre ellos. El diagrama de clases que modela este requerimiento es el siguiente, donde se observa que ambas clases se conocen entre sí:



En ambas clases, considerar que:

- El atributo estadoCivil, cuando se crea el objeto, toma por valor Soltero/a, al casarse Casado/a y al divorciarse Divorciado/a. Este cambio debe ser realizado desde los métodos correspondientes.
- El método divorcio(), además, coloca null en el atributo esposo/esposa, mientras que el método casarseCon() asigna el cónyuge indicado.
- El método datos() retorna: los datos del objeto (nombre, apellido y edad) ej:
María Gómez de 28 años (ejemplo Mujer)
- El método mostrarEstadoCivil () tiene la siguiente salida impresa:



María Gómez de 28 años - Casada

- El método casadaCon()/casadoCon(), tiene la siguiente salida impresa:
Pedro Leyes de 29 años está casado con María Gómez de 28 años (ejemplo Hombre)
- Implemente en Java el diagrama de clases, y luego una clase ejecutable RegistroCivil que emita el certificado de matrimonio.

EJERCICIO 3: Considerando el cambio en la ley nacional, que habilita el matrimonio igualitario:

Diseñe e implemente un diagrama de clases que permita emitir el certificado de matrimonio considerando las nuevas opciones.