Ejercicios
Herencia en Java

PARADIGMAS DE PROGRAMACIÓN I

Carlos Rojas Sánchez

Licenciatura en Informática

Universidad del Mar

Contenido

Ejercicio 1

Clase Cuenta con herencia

Desarrollar un programa que modele una cuenta bancaria que tiene los siguientes atributos, que deben ser de acceso protegido:

- Saldo, de tipo float.
- Número de consignaciones con valor inicial cero, de tipo int.
- Número de retiros con valor inicial cero, de tipo int.
- Tasa anual (porcentaje), de tipo float.
- Comisión mensual con valor inicial cero, de tipo float.

Clase Cuenta con herencia

La clase Cuenta tiene un constructor que inicializa los atributos saldo y tasa anual con valores pasados como parámetros. La clase Cuenta tiene los siguientes métodos:

- Consignar una cantidad de dinero en la cuenta actualizando su saldo.
- Retirar una cantidad de dinero en la cuenta actualizando su saldo. El valor a retirar no debe superar el saldo.
- Calcular el interés mensual de la cuenta y actualiza el saldo corres pondiente.
- Extracto mensual: actualiza el saldo restándole la comisión mensual y calculando el interés mensual correspondiente (invoca el método anterior).
- Imprimir: muestra en pantalla los valores de los atributos.

Clase Cuenta con herencia

La clase Cuenta tiene dos clases hijas:

- Cuenta de ahorros: posee un atributo para determinar si la cuenta de ahorros está activa (tipo boolean). Si el saldo es menor a $10 000, la cuenta está inactiva, en caso contrario se considera activa. Los siguientes métodos se redefinen:
  - Consignar: se puede consignar dinero si la cuenta está activa. Debe invocar al método heredado.
  - Retirar: es posible retirar dinero si la cuenta está activa. Debe invocar al método heredado.
  - Extracto mensual: si el número de retiros es mayor que 4, por cada retiro adicional, se cobra $1000 como comisión mensual. Al generar el extracto, se determina si la cuenta está activa o no con el saldo.

Clase Cuenta con herencia

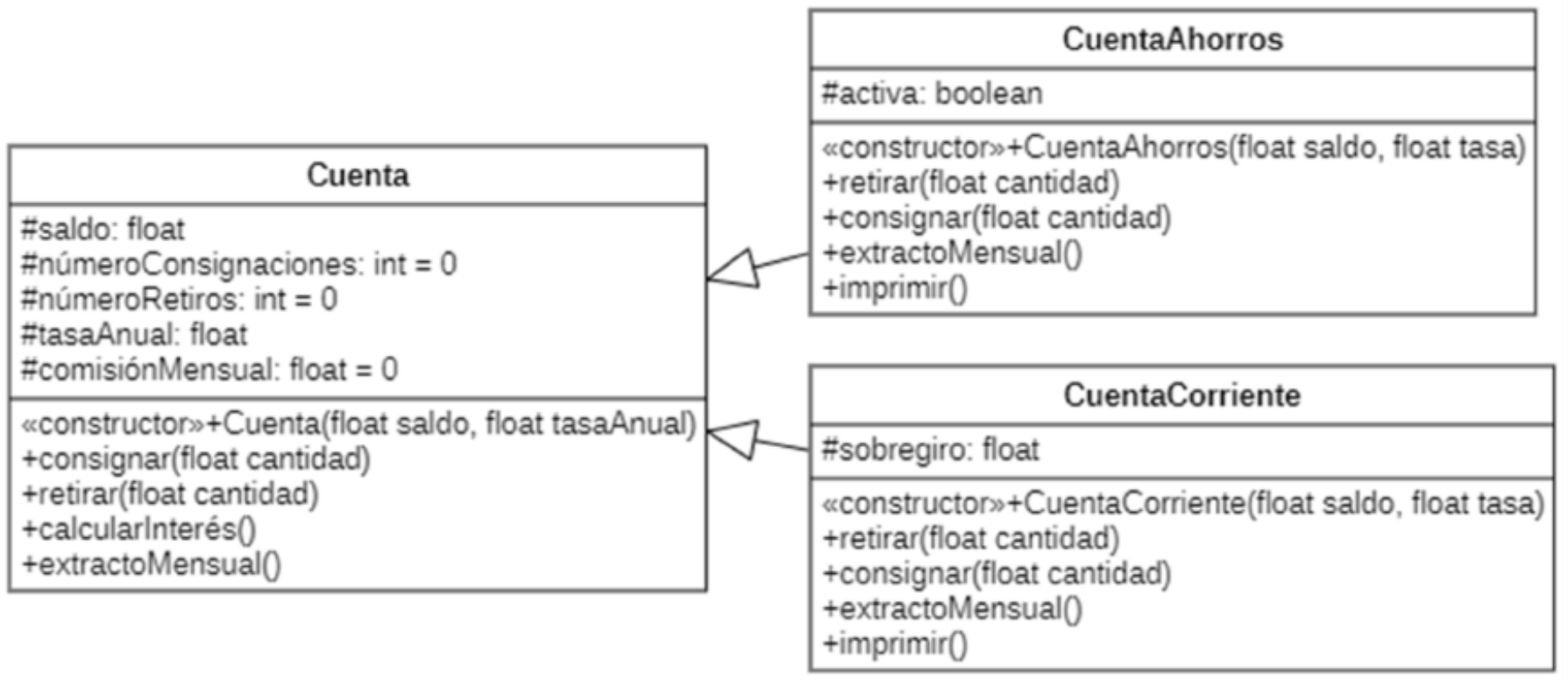
- Cuenta de Ahorros (continuación)
  - Un nuevo método imprimir que muestra en pantalla el saldo de la cuenta, la comisión mensual y el número de transacciones realizadas (suma de cantidad de consignaciones y retiros).
- Cuenta corriente: posee un atributo de sobregiro, el cual se inicializa en cero. Se redefinen los siguientes métodos:
  - Retirar: se retira dinero de la cuenta actualizando su saldo. Se pue de retirar dinero superior al saldo. El dinero que se debe queda como sobregiro.
  - Consignar: invoca al método heredado. Si hay sobregiro, la cantidad consignada reduce el sobregiro.

Clase Cuenta con herencia

- Cuenta de corriente (continuación)
  - Extracto mensual: invoca al método heredado.
  - Un nuevo método imprimir que muestra en pantalla el saldo de la cuenta, la comisión mensual, el número de transacciones realizadas (suma de cantidad de consignaciones y retiros) y el valor de sobregiro.

Realizar un método main que implemente un objeto Cuenta de ahorros y llame a los métodos correspondientes.

Clase Cuenta con herencia



Clase Cuenta con herencia

```
Cuenta de ahorros
Ingrese saldo inicial= $
100000
Ingrese tasa de interés= 0,10
Ingresar cantidad a consignar: $50000
Ingresar cantidad a retirar: $70000
Saldo = $ 80666.664
Comisión mensual =  $ 0.0
Número de transacciones = 2
```

Ejercicio II

Clase Inmueble con herencia

Se requiere realizar un programa que modele diferentes tipos de inmuebles. Cada inmueble tiene los siguientes atributos: identificador inmobiliario (tipo entero); área en metros cuadrados (tipo entero) y dirección (tipo String). Los inmuebles para vivienda pueden ser casas o apartamentos. Los inmuebles para vivienda tienen los siguientes atributos: número de habitaciones y número de baños. Las casas pueden ser casas rurales o casas urbanas, su atributo es la cantidad de pisos que poseen. Los atributos de casas rurales son la distancia a la cabecera municipal y la altitud sobre el nivel del mar. Las casas urbanas pueden estar en un conjunto cerrado o ser independientes.

Clase Inmueble con herencia

A su vez, las casas en conjunto cerrado tienen como atributo el valor de la administración y si incluyen o no áreas comunes como piscinas y campos deportivos. De otro lado, los apartamentos pueden ser apartaestudios o apartamentos familiares. Los apartamentos pagan un valor de administración, mientras que los apartaestudios tienen una sola habitación.

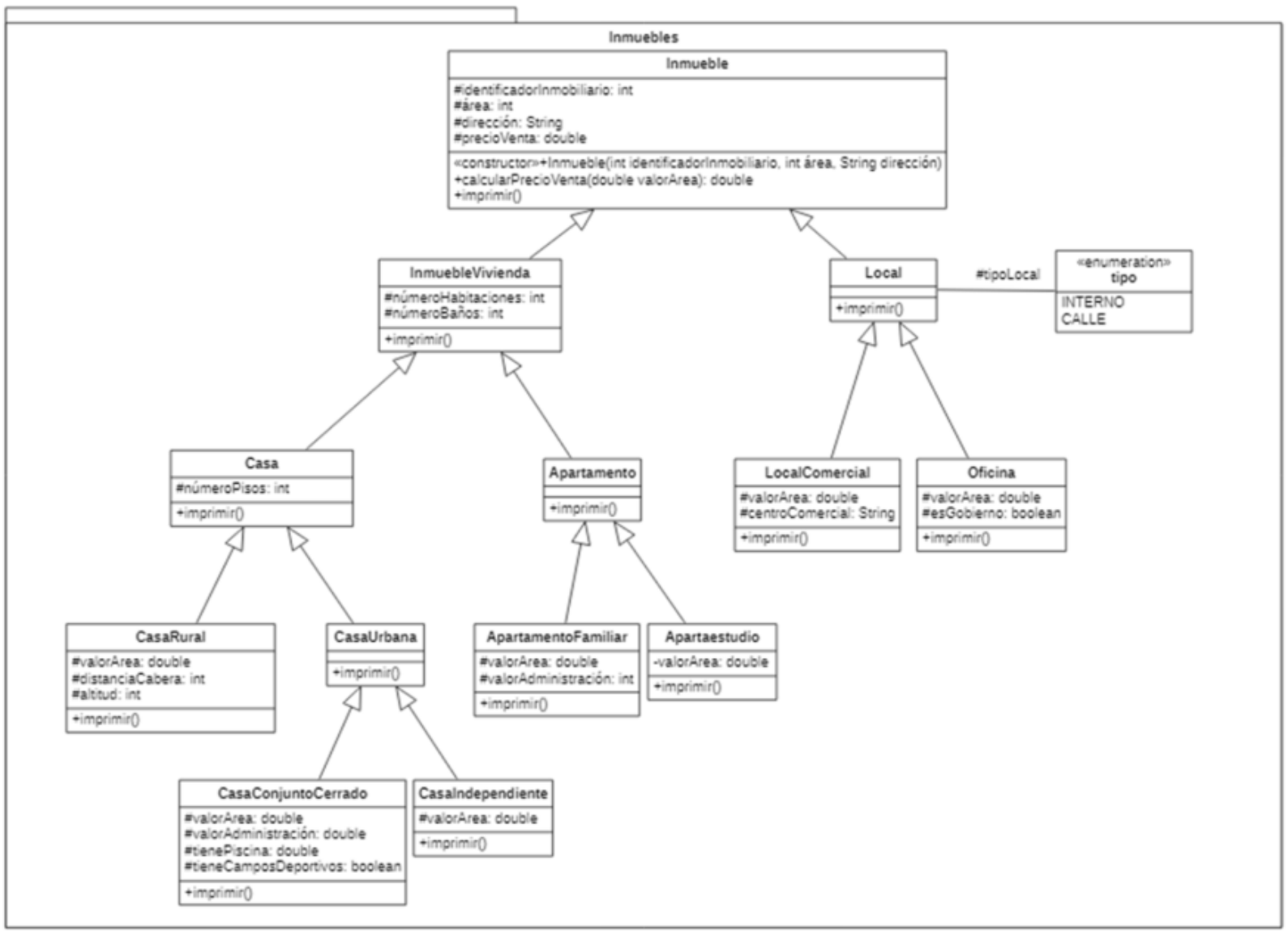
Clase Inmueble con herencia

Los locales se clasifican en locales comerciales y oficinas. Los locales tienen como atributo su localización (si es interno o da a la calle). Los locales comerciales tienen un atributo para conocer el centro comercial donde están establecidos. Las oficinas tienen como atributo un valor boolean para determinar si son del Gobierno. Cada inmueble tiene un valor de compra. Este depende del área de cada inmueble según la tabla siguiente:

Clase Inmueble con herencia

Inmueble	Valor por metro cuadrado
Casa rural	$ 1 500 000
Casa en conjunto cerrado	$ 2 500 000
Casa independiente	$ 3 000 000
Apartaestudio	$ 1 500 000
Apartamento familiar	$ 2 000 000
Local comercial	$ 3 000 000
Oficina	$ 3 500 000

Clase Inmueble con herencia



Clase Inmueble con herencia

```
Datos apartamento
Identificador inmobiliario = 103067
Area = 120
Dirección = Avenida Santander 45-45
Precio de venta = $2.4E8
Número de habitaciones = 3
Número de baños = 2
Valor de la administración = $200000

Datos apartamento
Identificador inmobiliario = 12354
Area = 50
Dirección = Avenida Caracas 30-15
Precio de venta = $7.5E7
Número de habitaciones = 1
Número de baños = 1
```

Ejercicio III

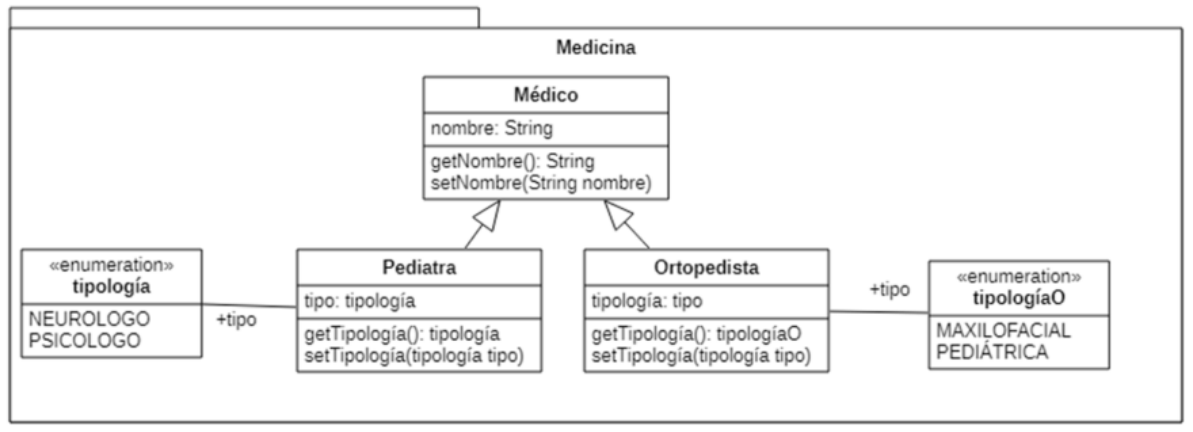
Clase Médico con jerarquía de herencia

Un médico posee como atributo su nombre y métodos get y set de dicho atributo. Los pediatras y ortopedistas son dos tipos de médicos. Estas dos subclases se modelan como subclases de médico. Los pediatras pueden ser pediatras neurólogos o pediatras psicológicos. Esta tipología se modela como un atributo enumerado de pediatra. De igual manera, los ortopedistas cuentan con un atributo enumerado para describir su tipo que puede ser maxilofacial o pediátrica. Estas subclases cuentan con métodos get y set.

Clase Médico con jerarquía de herencia

En una clase de prueba, en su método main, se solicita crear un vector con objetos tanto de tipo pediatra como ortopedistas. Luego, en pantalla se debe mostrar qué tipo de objeto está ubicado en cada posición del vector. Las clases deben estar contenidas en un paquete denominado Medicina.

Clase Médico con jerarquía de herencia



Clase Médico con jerarquía de herencia

```
El objeto en el indice 0 es de la clase Médico
El objeto en el indice 1 es de la clase Ortopedista
El objeto en el indice 2 es de la clase Pediatra
```

Ejercicio IV

Jerarquía de Herencia de Mamífero con Interfaz

Hacer un programa que implemente las siguientes clases y métodos relacionados con una jerarquía taxonómica de animales.

- Los mamíferos son una clase abstracta con el método abstracto amamantar crías, que no devuelve nada.
- Las ballenas son mamíferos e implementan el método abstracto heredado, la pantalla muestra un mensaje indicando que ellas amamantan a sus crías.
- Los animales ovíparos son una interfaz con el método poner huevos.

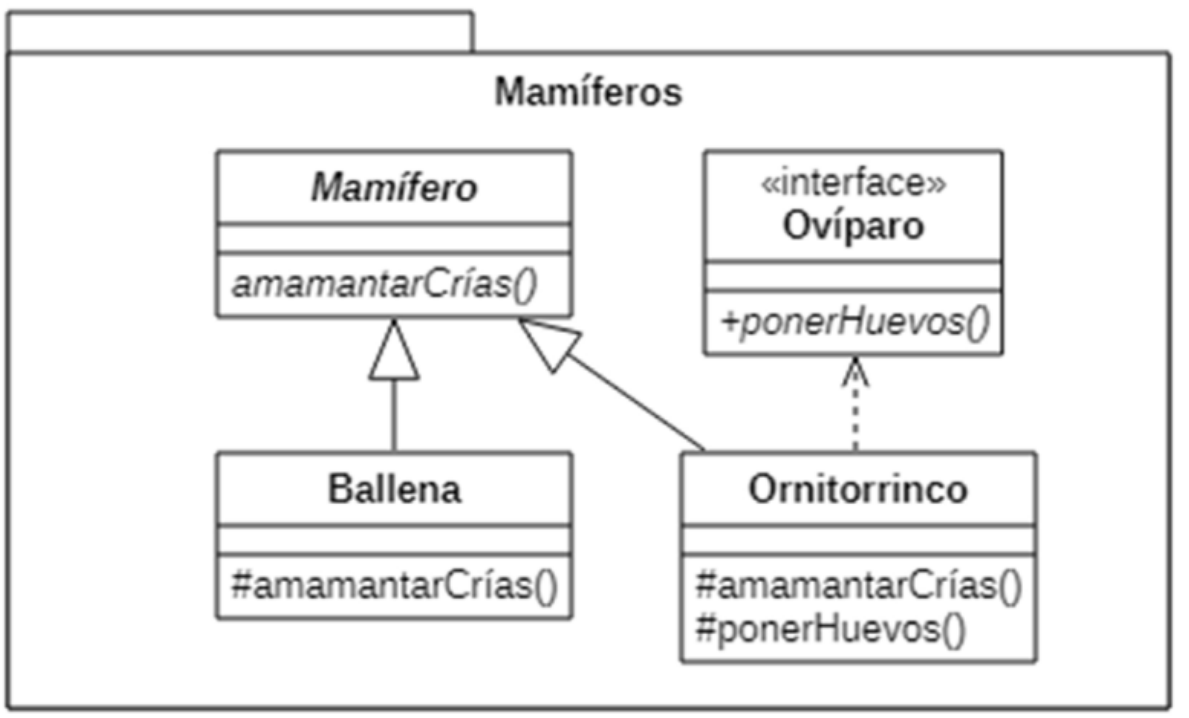
Jerarquía de Herencia de Mamífero con Interfaz

Continuación.

- El ornitorrinco es un mamífero que pone huevos. Por lo tanto, es una clase que hereda de mamífero e implementa la interfaz Ovíparo. El método heredado de la clase padre muestra en pantalla que el ornitorrinco amamanta a sus crías, y el método implementado desde la interfaz muestra en pantalla que pone huevos.

Generar un método main donde se crean una ballena y un ornitorrinco y se invocan los métodos heredados e implementados.

Jerarquía de Herencia de Mamífero con Interfaz



Jerarquía de Herencia de Mamífero con Interfaz

```
La ballena amamanta a sus crías.
El ornitorrinco amamanta a sus crías.
El ornitorrinco pone huevos.
```

Ejercicio V

Clase Vehículo con Jerarquía de Herencia e Implementación de Varias Interfaces

Un vehículo posee una velocidad actual y una velocidad máxima (ambas en km/h). Un vehículo tiene dos métodos abstractos:

- El método acelerar permite incrementar la velocidad actual su mándole la velocidad pasada como parámetro. La velocidad actualizada no puede superar la velocidad máxima.
- El método frenar permite disminuir la velocidad actual restándole la velocidad pasada como parámetro. La velocidad actualizada no puede ser negativa.
- El método imprimir muestra en pantalla la velocidad actual y la velocidad máxima del vehículo.

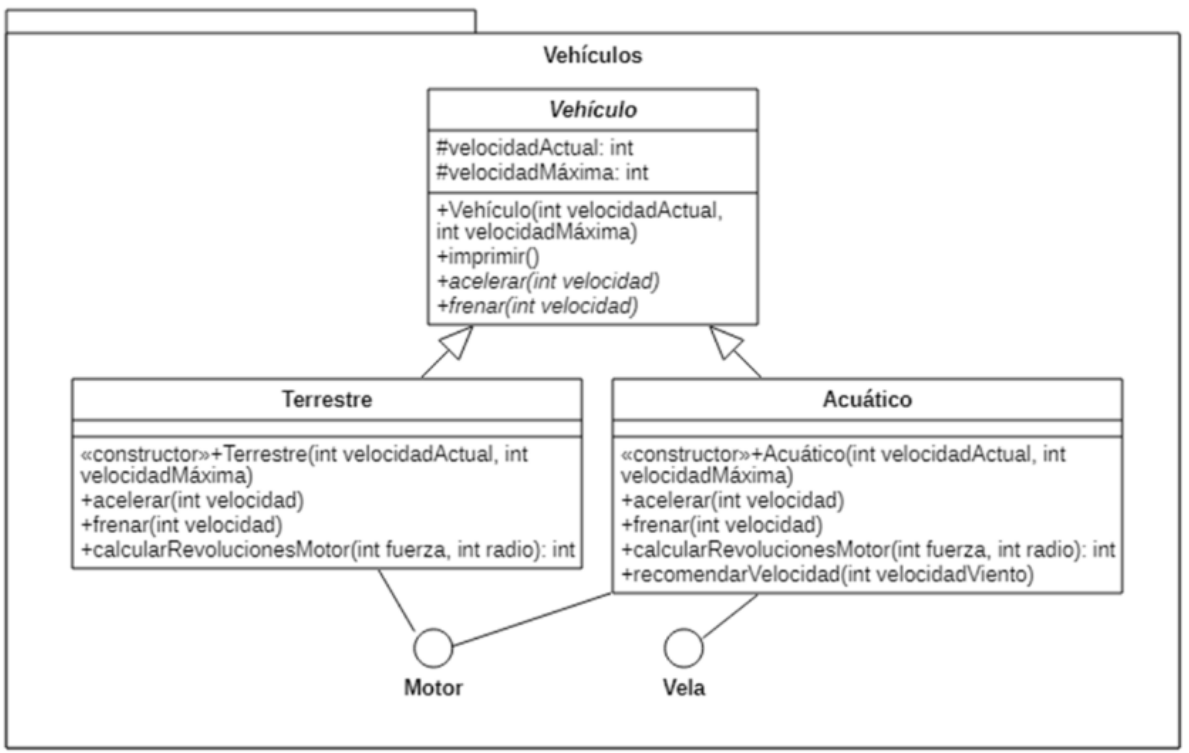
Clase Vehículo con Jerarquía de Herencia e Implementación de Varias Interfaces

- Existen dos tipos de vehículos: vehículos terrestres y acuáticos. En primer lugar, los vehículos terrestres tienen como nuevos atributos: la cantidad de llantas y el uso del vehículo (militar o civil). En segundo lugar, los vehículos acuáticos tienen como nuevos atributos su tipo (de superficie o submarino) y capacidad de pasajeros.
- A su vez, existen dos interfaces: Motor y Vela. La clase Vehículo terrestre implementa la interfaz Motor. La clase vehículo acuático implementa la interfaz Vela.

Clase Vehículo con Jerarquía de Herencia e Implementación de Varias Interfaces

- La interfaz Motor tiene el siguiente método: calcularRevolucionesMotor(int fuerza, int radio): el número de revoluciones se calcula como la multiplicación de la fuerza del motor por su radio.
- La interfaz Vela tiene los siguientes métodos: recomendarVelocidad(int velocidadViento): si la velocidad del viento es mayor a 80 km/h es muy alta, se recomienda no salir a navegar y, por lo tanto, la velocidad actual debe ser cero. Si la velocidad del viento es menor a 10 km/h es muy baja y tampoco se recomienda salir a navegar.

Clase Vehículo con Jerarquía de Herencia e Implementación de Varias Interfaces



Clase Vehículo con Jerarquía de Herencia e Implementación de Varias Interfaces

```
Camioneta
Velocidad actual = 100
Velocidad máxima = 250
Nueva Velocidad actual= 150
Moto acuática
Velocidad actual = 50
Velocidad máxima = 110
Revoluciones del motor = 2400
```

Ejercicio VI

Interfaz MatchDeportivo y sus Interfaces Heredadas

Se requiere desarrollar un programa que defina una interfaz denominada MatchDeportivo que represente un encuentro entre dos equipos deportivos. La interfaz cuenta con dos métodos: setEquipoLocal y setEquipoVi sitante que no retornan nada, pero cada uno recibe como parámetro un String con el nombre del equipo. La interfaz posee dos interfaces heredadas:

- Partido de fútbol: define dos nuevos métodos, setGolesEquipoLo cal(int marcador) que permite asignar la cantidad de goles que realiza un equipo de fútbol que juega como local y setGolesEquipoVi sitante(int marcador) para asignar la cantidad de goles que realiza un equipo de fútbol que juega como visitante. Ambos métodos no devuelven valores. La interfaz cuenta con un atributo que identifica la duración de un partido de fútbol, la cual tiene un valor de 90 minutos.

Interfaz MatchDeportivo y sus Interfaces Heredadas

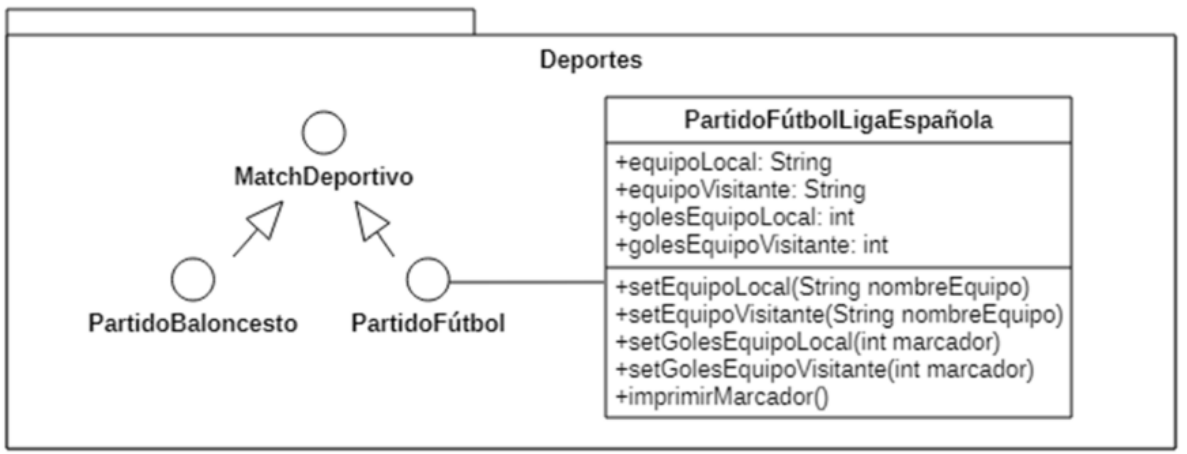
Continuación:

- Partido de baloncesto: define dos nuevos métodos, setCestasEqui poLocal(int marcador) que permite asignar la cantidad de cestas que realiza un equipo de baloncesto que juega como local y set CestasEquipoVisitante(int marcador) que permite asignar la cantidad de cestas que realiza un equipo de baloncesto que juega como visitante. Ambos métodos no devuelven valores. La interfaz cuenta con un atributo que identifica la duración de un partido de baloncesto, la cual tiene un valor de 40 minutos.

Interfaz MatchDeportivo y sus Interfaces Heredadas

Se debe definir una clase PartidoFútbolLigaEspañola que implementa la interfaz Partido de fútbol, tiene los atributos: equipoLocal y equipoVisitante (ambos de tipo String) y golesEquipoLocal y golesEquipoVisitante (ambos de tipo int). Además, cuenta con métodos get y set para cada atributo y un método imprimirMarcador que imprime el marcador obtenido en el partido por los dos equipos.

Interfaz MatchDeportivo y sus Interfaces Heredadas



Interfaz MatchDeportivo y sus Interfaces Heredadas

```
Duración del partido =90
Equipo local: Real Madrid: 3 - Equipo visitante: Barcelona: 3
```