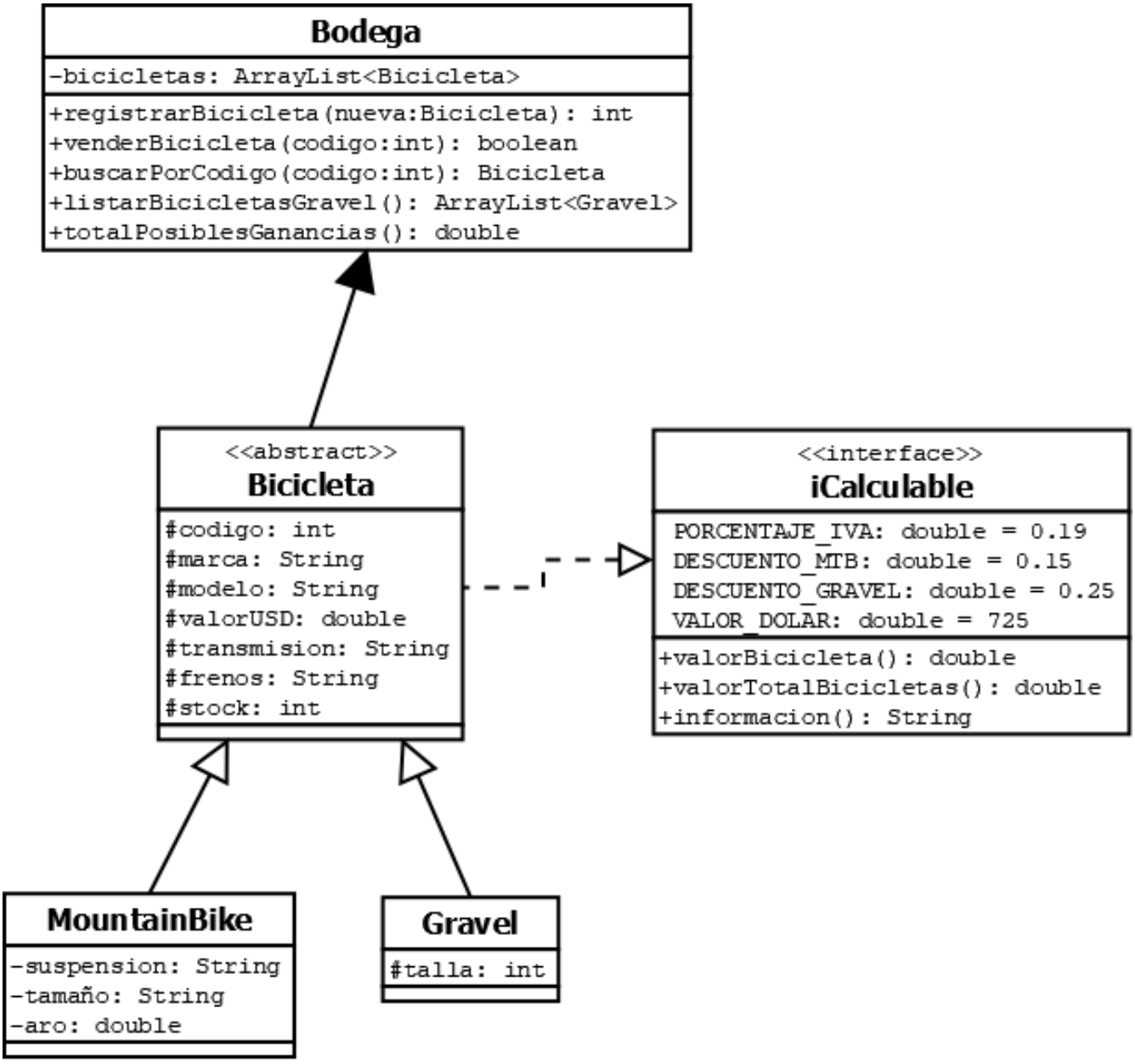
Docente: Christian Sarmiento Zampillo

Ejercicio Practico Nro. 05

POO – Herencia, Clases Abstractas, Interface, Colecciones

A partir del siguiente Diagrama de Clases, codificar las siguientes clases y su estructura.



Docente: Christian Sarmiento Zampillo

Para el siguiente ejercicio se necesita que codifique las siguientes clases según los siguientes requerimientos:

Cada clase debe tener sus respectivos constructores, sobrecargando los constructores correspondientes.

Generar accesadores y mutadores para todas las clases.

La clase MountainBike no debe permitir a partir de ella más herencia.

Reglas de negocio:

- El largo de la Marca y modelo deben ser siempre mayor a 0.
- El tamaño debe ser XS, S, M, L y XL
- La talla debe estar entre 44 y 58

La implementación de los métodos de la *interface* debe ser según se indica a continuación:

valorBicicleta: este método retorna el valor de la bicicleta en pesos chilenos, debe considerar el tipo de bicicleta para aplicar el descuento correspondiente. Además, debe obtener los impuestos y considerar el valor dólar que se plantea en el ejercicio.

valorTotalBicicletas: este método retorna el monto total que cuestan todas las bicicletas según el valor y el stock.

informacion: este método debe entregar la siguiente información:

MountainBike: Cannondale Catalyst | Suspensión Delantera | Transmisión 3x9 | Frenos Disco Hidráulico | Tamaño M - Aro 27.5 | Valor: $ 479.500

Gravel: Norco Search 105 | Talla 50.5 | Transmisión 2x11 | Frenos Disco Mecánico | Valor: $1.199.900

En la clase Bodega se deben codificar los siguientes métodos:

registrarBicicleta: este método deberá retornar un número entero. Retornará 0 si la bicicleta no existe en los registros, se debe agregar a la colección. Retornará 1 si la bicicleta ya existe en la colección. El stock de bicicletas existentes se debe actualizar sumando la cantidad de bicicletas que se quieren registrar. Finalmente retornará -1 si sucedió un error.

venderBicicleta: este método retornará un valor TRUE si al momento de vender una bicicleta esta existe y además tenemos stock (mayor a 0). En caso contrario retornar FALSE.

buscarPorCodigo: este método debe retornar una bicicleta la cual debe ser buscada por su código.

listarBicicletasGravel: este método debe retornar una colección con todas las bicicletas Gravel que hay en la colección principal.

totalPosiblesGanancias: este método debe retornar el total de ganancias que generaría la venta de todas las bicicletas que hay en bodega según su stock.

Para probar el funcionamiento de su código se requiere que en la clase Main realice las siguientes operaciones

1. Crear una Bodega.
2. Crear al menos 10 bicicletas. Considere en el proceso de creación de las Bicicletas, crear instancias con los mismos datos para ver el proceso de actualización de stock.
3. Registre las bicicletas en la bodega.
4. Ejecutar los métodos que permiten registrar, buscar, vender, listar y mostrar el total.
5. Recuerde documentar todo su código mediante el uso de JAVADOC y general la documentación del proyecto.

Utilice la nomenclatura del lenguaje para la definición de clases, atributos y encapsule todo su código.