

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



LABORATORIO DE INF-121

Guia de Ejercicios

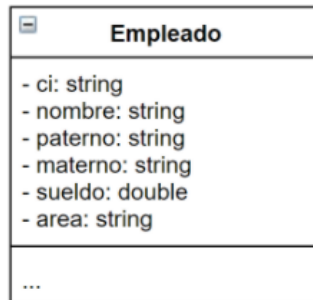
Auxiliares:

Par.	Docente	Auxiliar	Horario	Celular
A	Lic. Tarquino	Univ. David Josue Meneces Zelaya	Sabado 08:00 a 10:00	76279697
B	Lic. Tarquino	Univ. Rodrigo Ticona Coronel	-	61120416
C y D	Lic. Huanca	Univ. Noemi Silvia Guarachi Vasquez	Viernes 10:00 a 12:00	73055731
E y F	Lic. Aruquipa	Univ. Sergio Alejandro Pauca-ra Saca	Viernes 08:00 a 10:00	76236361

GESTIÓN I/2022

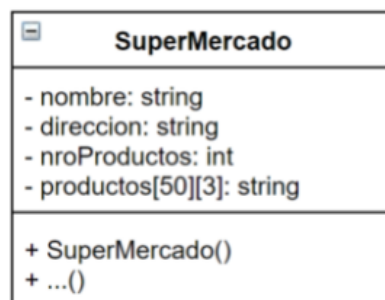
Programacion Orientada a Objetos

1. La empresa ABC, tiene los siguientes datos de sus empleados:



Se pide:

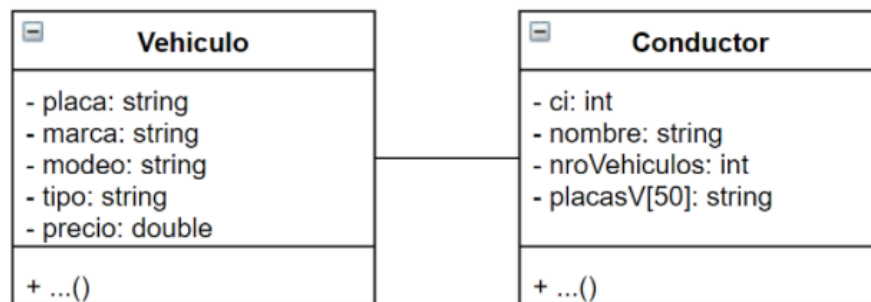
- Instanciar 2 empleados de manera diferente, y mostrarlos.
 - De los 2 empleados mostrar al empleado que tiene mayor sueldo.
 - Modificar el área de trabajo de un empleado con el área X.
 - Verificar si ambos empleados son del mismo área.
2. Sea el siguiente diagrama:



Dónde: `producto[i][0]` = nombre, `producto[i][1]` = stock, `producto[i][2]` = precio.
Se pide:

- Instanciar 2 supermercados de forma distinta.
- Mostrar los 2 supermercados.
- Mostrar el supermercado con más stock disponible del producto de nombre X.
- Mostrar el nombre de los productos que tiene el supermercado de nombre Y, que son más baratos en el supermercado de nombre X.

- e) Implementar un método para verificar si el supermercado tiene stock disponible del producto de nombre X.
3. Se tiene el registro de los vehículos por conductor en el siguiente diagrama:

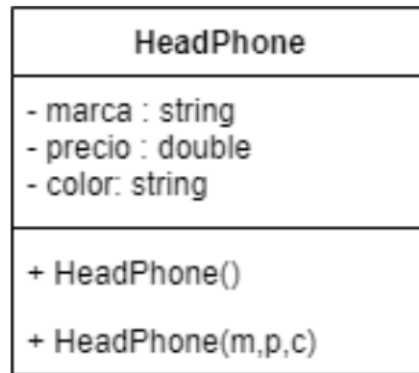


Donde: `tipo = {automóvil, camioneta, bus, motocicleta, ...}`, `modelo = {Corolla, Prius, ...}`, `marca = {Toyota, Nissan, ...}`

- a) Instanciar 3 Vehículos y 2 Conductores.
- b) Mostrar los Vehículos del Conductor con CI X.
- c) Mostrar el Conductor del Vehículo con placa X.
- d) Verificar si el Conductor con CI X tiene algún Vehículo de tipo “Camioneta”.

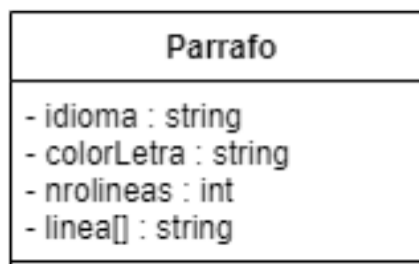
Polimorfismo

1. Sea el siguiente diagrama:



Se pide:

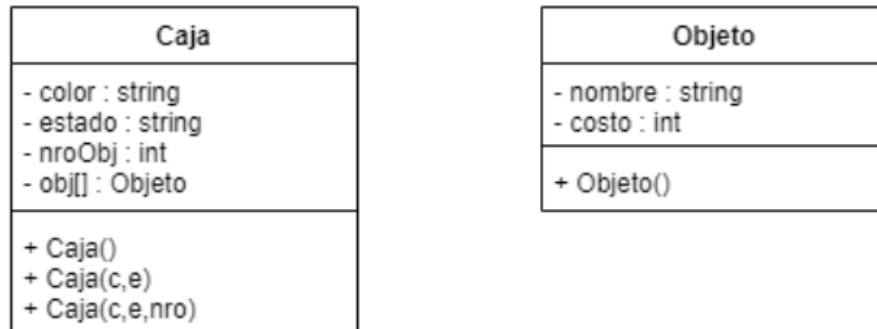
- Implementar 2 Headphones usando distintos constructores
 - Sobrecargar el operador (++) para leer los datos del HeadPhone
 - Sobrecargar el operador (- -) para mostrar los datos del HeadPhone
 - Sobrecargar el operador (+) para aumentar el precio del HeadPhone X con un monto Y.
2. Sea el siguiente diagrama de clases:



Se pide:

- Instanciar 2 Párrafos de forma distinta y mostrarlos.
- Sobrecargar el método “mostrar” para mostrar las lineas en el rango [a,b] donde a,b son ingresados por teclado
- Sobrecargar al operador ! (true/false) para verificar si el párrafo tiene texto.
- Sobrecargar al operador (++) para contar el número de palabras que hay en el párrafo.

3. Sea el siguiente diagrama de clase:

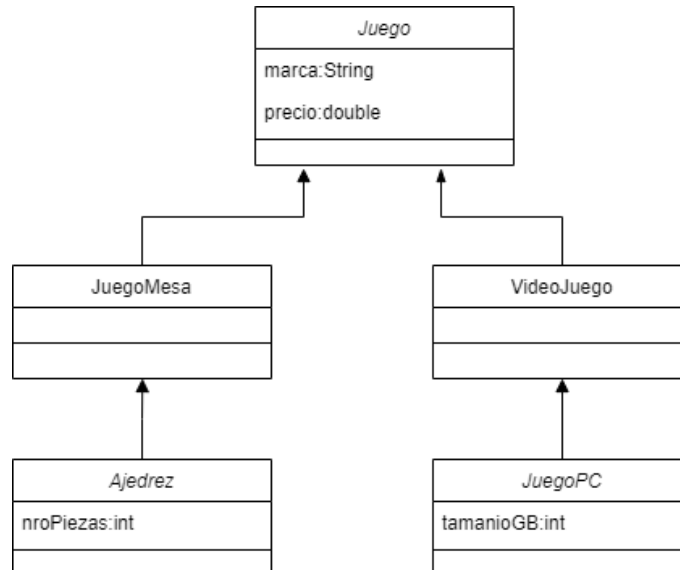


Se pide:

- Instanciar 3 Cajas de forma distintas y mostrarlas
- Sobrecargar el operador (- -) en la clase Objeto, para disminuir el costo de todos los objetos de la caja X en un 50 %.
- Sobrecargar el operador (-) para vaciar el contenido de una caja X que se encuentra en mal estado hacia una caja Y que está en buen estado
- Sobrecargar el operador (- -) en la clase Caja, para leer un nuevo color para la caja, un nuevo estado para la caja, y mostrar los objetos que tienen nombres palindromes.

Herencia

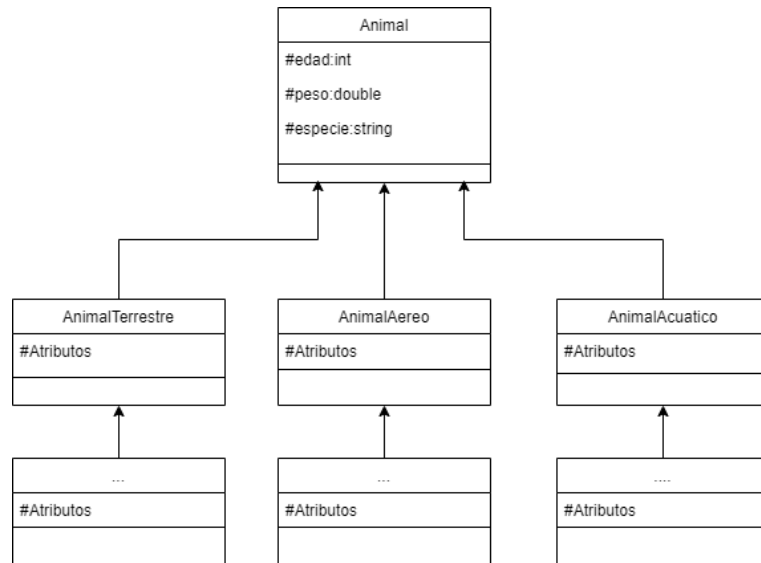
1. Sea el siguiente diagrama de clases.



Se pide:

- Implementar el diagrama de clases, con constructores por defecto, getters y setters.
- Instanciar 1 objeto **Ajedrez** y 1 objeto **JuegoPC**.
- Mostrar el objeto mas caro.
- Mostrar el objeto con el menor nombre lexicograficamente.

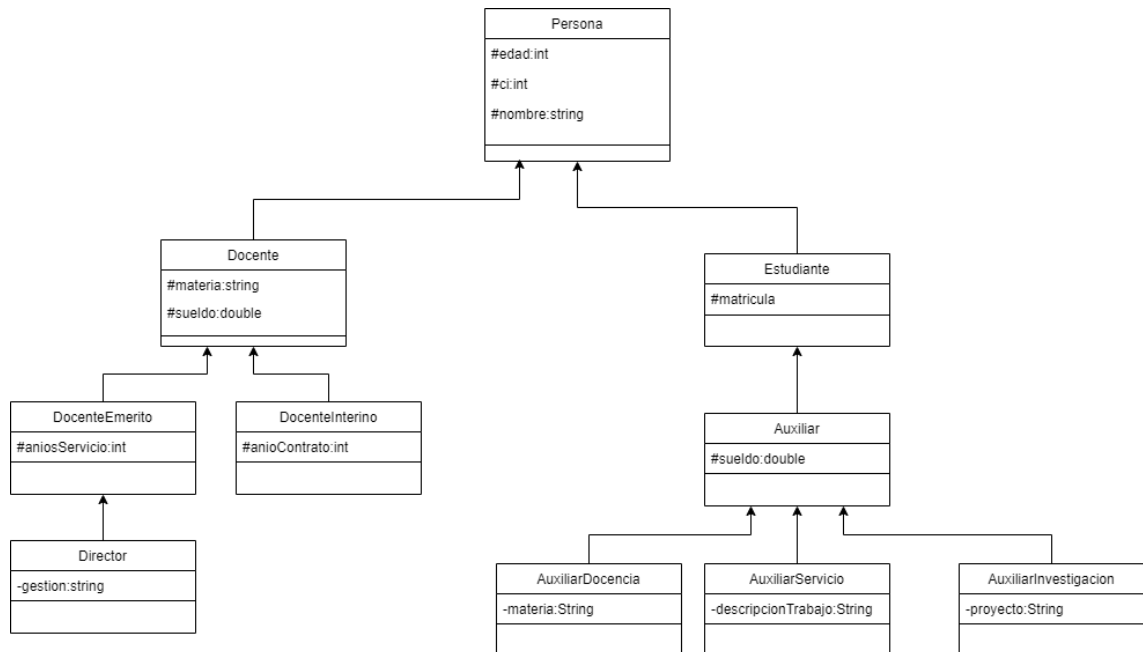
2. Sean el siguiente diagrama de clases:



Se pide:

- Completar el diagrama de clases con al menos 2 atributos y 2 métodos no comunes, además de dar un nombre a las clases hijo que no tienen un nombre asignado
- Instanciar 2 objetos de cada hijo de **AnimalTerrestre**, **AnimalAcuático** y **AnimalAéreo**.
- De los 6 objetos instanciados, mostrar cuales son los animales donde sus nombres empiezan con la letra **X**.
- De los 6 objetos instanciados, mostrar cuales son los animales que tienen la menor edad (sin usar vectores).
- De los 6 objetos instanciados mostrar los animales con mayor peso. (sin usar vectores).

3. Sean el siguiente diagrama de clases:

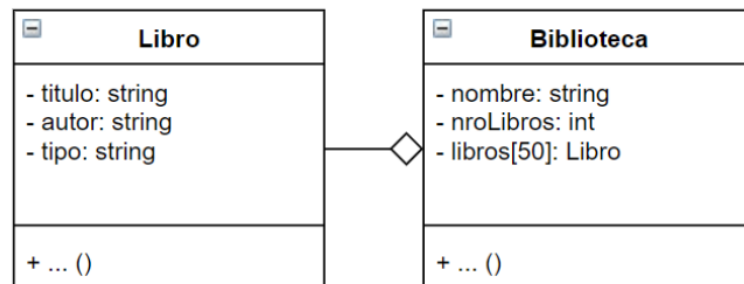


Se pide:

- a) Implementar todo el diagrama de clases e instanciar:
 - 1 objeto Director
 - 2 objetos DocenteEmerito
 - 2 objetos DocenteInterino
 - 1 objeto AuxiliarDocencia
 - 1 objeto AuxiliarServicio
 - 1 objeto AuxiliarInvestigacion
 - 2 objetos Estudiante
- b) De entre los Docentes y Auxiliares hallar el docente y auxiliar que tiene mayor sueldo.
- c) Mostrar la materia que da el Director.
- d) Hallar al docente con menor edad.
- e) Mostrar a los DocentesEmerito donde su carnet termina en un numero par.

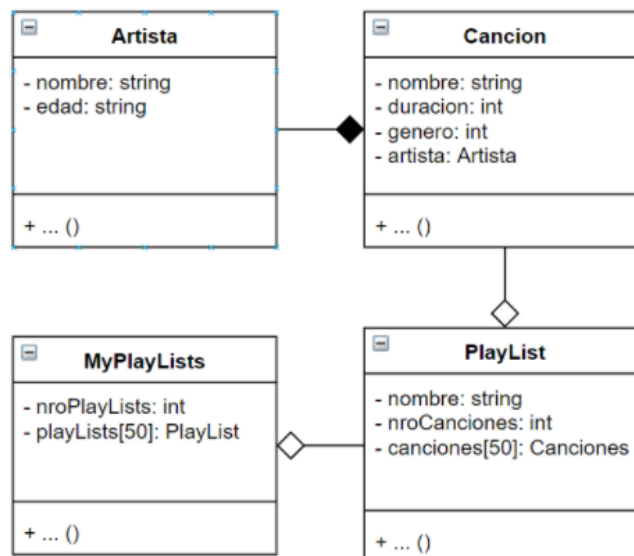
Agregación y Composición

1. Sea el siguiente diagrama:



Se pide:

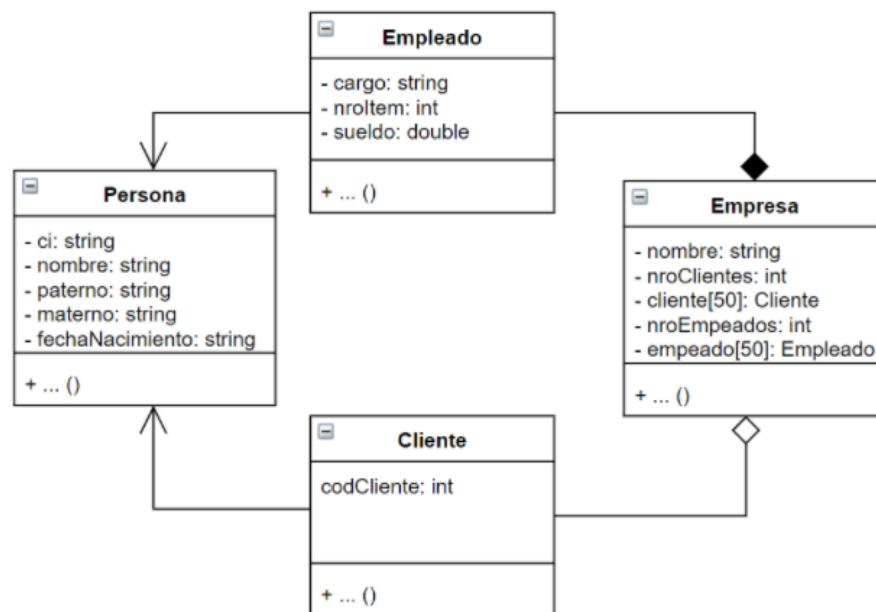
- Implementar todas las clases y sus métodos principales e instanciar 2 Bibliotecas.
 - Verificar si el libro con título X está en la biblioteca.
 - Mostrar los libros que se encuentren en ambas bibliotecas.
2. Dado el siguiente diagrama:



Se pide:

- Implementar todas las clases y sus métodos principales.
- Mostrar la(s) Playlist(s) que tiene(n) más canciones.

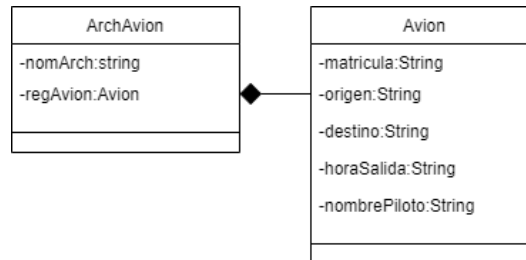
- c) Verificar si tenemos alguna canción del Artista de nombre X.
 - d) Mostrar la cantidad de canciones con género X de cada una de las PlayList.
 - e) Ordenar cada Playlist por duración.
 - f) Eliminar todas las canciones que son del Artista con nombre X.
3. Sea el siguiente diagrama:



- a) Implementar todas las clases y sus métodos principales.
- b) Eliminar al empleado con CI X de la lista de empleados de la empresa.
- c) Ordenar a todos los empleados según su salario.
- d) Eliminar al(los) empleado(s) que es(son) cliente y empleado en la empresa al mismo tiempo.

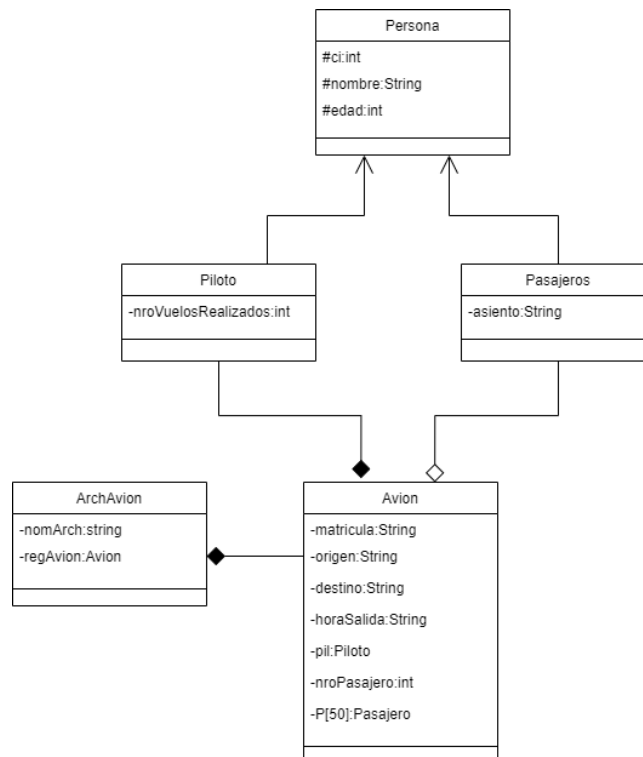
Persistencia

1. Dado el siguiente diagrama de clases:



- Crear el archivo
- Adicionar 3 objetos Avion al archivo.
- Listar el archivo.
- Eliminar al avion con matricula **M**.

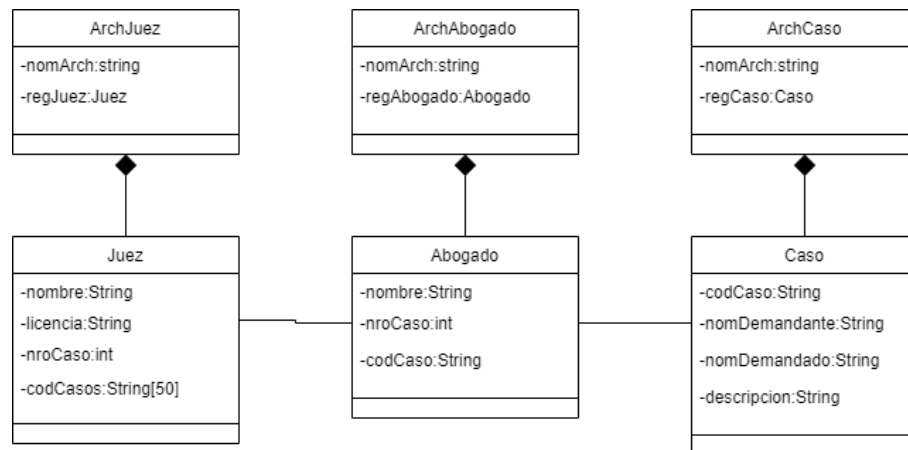
2. Dado el siguiente diagrama de clases:



- Crear el archivo
- Adicionar 3 objetos Avion al archivo.

- c) Listar el archivo.
- d) Adicionar al Pasajero **W**, al avion con matricula **X**.
- e) Mostrar los pasajeros del avion, cuyo piloto tiene el nombre **Y**.
- f) Eliminar el Avion con destino **X**.

3. Sea el siguiente diagrama de clases.

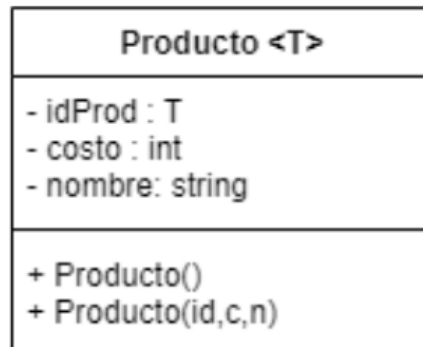


Se pide:

- a) Implementar el diagrama de clases, por cada archivo crear 2 objetos.
- b) Mostrar el nombre del demandante del caso que maneja el Juez con nombre **X**.
- c) Eliminar los casos que maneja el Abogado con nombre **Y**.
- d) Mostrar los nombres de los demandados y demandantes de los casos que maneja el juez con licencia **Z**.

Genericidad

1. Sea el siguiente diagrama de clases, donde $T = \{\text{string}, \text{int}\}$:



Se pide:

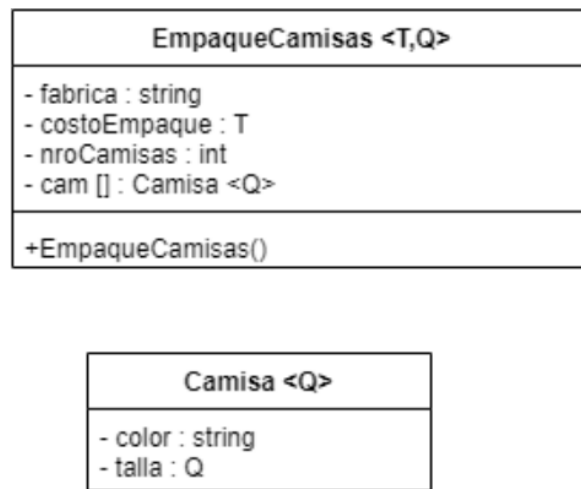
- Instanciar un producto con idprod de tipo string (ej: "Jabon-123") y mostrarlo
 - Instanciar un producto con idprod de tipo int (ej : "123456") y mostrarlo
 - Crear un método que compare dos productos del mismo tipo y muestre el producto mas caro
 - Sobrecargar al operador (+) para aumentar el precio del producto X de cualquier tipo con un monto Y.
2. Sea el siguiente diagrama de clases, donde $T = \{\text{string}, \text{int}\}$, $Q = [\text{int}, \text{float}, \text{double}]$, $R = \{\text{int}, \text{char}, \text{string}\}$



Se pide:

- Instanciar 3 tenis de diferentes tipos y mostrarlos.
- Crear el metodo aumento(x) que permite aumentar el precio del tenis en un monto x

- c) Crear un vector de Tenis de tipo $T = \text{string}$, $Q = \text{double}$, $R = \text{int}$ con 5 tenis y mostrarlo.
 - d) Disminuir el costo de los Tenis de Talla 41 en un 20
 - e) Ordenar el vector de Tenis por su costo y mostrarlos
3. Sea el siguiente diagrama de clases, donde $T = \{\text{int}, \text{float}, \text{double}\}$, $Q = \{\text{int}, \text{string}\}$



Se pide:

- a) Instanciar 2 empaques de distinto tipo y mostrarlas
- b) Instanciar 3 camisas de distinto tipo y mostrarlas
- c) Agregar las camisas instanciadas en el inciso c) y agregarlos a sus empaques correspondientes y mostrar los empaques
- d) Mostrar la cantidad de camisas de color “azul” en un objeto empaquecamisas
- e) Aumentar el costo del empaque cuyo número de camisas sea número primo.