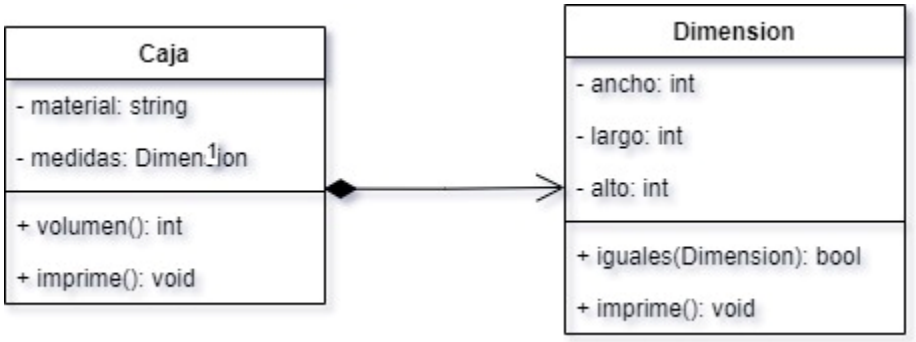
"De Acuerdo con el Código de Honor del estudiante del Tecnológico de Monterrey, mi actuación en este examen estará guiada por la integridad académica."

(55 puntos) Implementa en C++ las clases mostradas en el diagrama UML de abajo. Revisa en el apartado **Especificaciones**, algunos requerimientos que deberás considerar en la implementación. Empieza por la clase **Dimension**.



Especificaciones para tomar en cuenta:

Las dos clases deben tener la definición en un archivo hpp y las implementaciones en un archivo cpp.

CLASE DIMENSION:
Representa las dimensiones de algún objeto tridimensional, los atributos son enteros, **no se deben permitir medidas negativas, la medida más pequeña que puede recibir es 2**. Por lo que **deberás validar** que, si te pasan medidas menores a 2, la medida se establecerá por default a la medida mínima. TIP: valida en los setters y usa los setters en el constructor (ya tienes el constructor con parámetros, no lo modifiques).

1. Implementa el **constructor default** Este constructor inicializa todos los atributos en 2.
2. **Implementa los getters** de todos los atributos.
3. **Implementa los setters** de todos los atributos, en los setters deberás hacer la validación, si el valor recibido como parámetro es menor que 2, le asignas por default un 2, de lo contrario, asignas el valor recibido como parámetro.
4. El método **imprime()** despliega a pantalla todos los atributos del objeto, utiliza el formato que tú quieras para esto (dale personalidad a tu imprime).
5. El método **iguales(Dimension&)**, devuelve un entero 1 si ***"ésta"*** Dimensión es igual a la que ***recibe como parámetro***. Decimos que son iguales, si coinciden en todas las medidas. Regresa ***0*** en caso contrario.

CLASE CAJA:
Representa una caja que puede estar hecha de diferente material. Es capaz de calcular el volumen que puede guardar.

1. Observa que el atributo medidas es un **objeto de la clase Dimension**.
2. Codifica los **dos constructores (por defecto y con parámetros), así como getters y setters** para todos los atributos. El material por defecto es ***"carton".***

3. El método **imprime()** imprime todos los datos de la caja, incluyendo las medidas de la misma. Utiliza el formato que tú quieras para esto (dale personalidad a tu imprime). y recuerda que tienes un método en Dimensión que imprime en el formato que tu diseñaste.
4. El método **volumen()** regresa un entero resultado del cálculo del volumen de la caja.

(15 puntos) En el programa principal exercise.cpp prueba tus clases de la siguiente manera:

- En la función main() tienes declarada una constante llamada NUM que te servirá para determinar el tamaño de un arreglo.
- **(1 punto)** En la función main(), **declara un arreglo** de tipo **Caja** de tamaño NUM.
- **(3 puntos) Completa la función llenaArreglo,** la cual sirve para "cambiar todos los datos de los objetos" del arreglo con datos que recibe del usuario. No olvides que el atributo ***medidas*** es un objeto de la clase Dimension.
- **(3 puntos) Implementa la función imprimeCajas** para desplegar a pantalla TODOS los objetos Caja del arreglo.
- **(2 puntos)** En la función main(), después de crear el arreglo, llama a la función **llenaArreglo** con el arreglo recién creado y la constante que tiene el tamaño del arreglo.
- **(2 puntos)** Posteriormente llama a la función **imprimeCajas** para que se desplieguen a pantalla los datos de las cajas.
- **(2 puntos)** Escribe las instrucciones necesarias para imprimir a pantalla las medidas de la última caja del arreglo. Y despliega también su volumen.
- **(2 puntos)** Escribe las instrucciones necesarias para determinar con un mensaje si la primera caja del arreglo y la última caja del arreglo tienen medidas iguales (usa el método de la clase Dimension adecuado)

NOTA: Son 10 pruebas con 20 asserts que revisa el autograder y que debes pasar para obtener los puntos de la implementación de las clases. Lo del programa principal lo califico de manera manual.