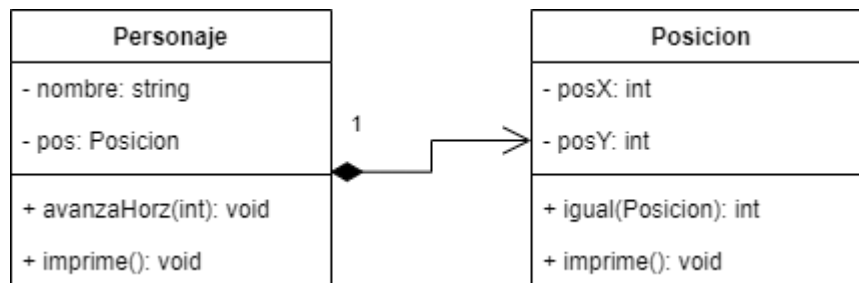


“De Acuerdo con el Código de Honor del estudiante del Tecnológico de Monterrey, mi actuación en este examen estará guiada por la integridad académica.”

(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 **Posicion**.



Especificaciones para tomar en cuenta:

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

CLASE POSICION:

Representa la posición dentro de una pantalla, es por esto que los datos son enteros, las medidas son en pixeles. Tampoco tendremos posiciones negativas, la posición 0,0 de una pantalla es generalmente la esquina superior izquierda. Las coordenadas x aumentan a la derecha. Las coordenadas y aumentan de arriba a abajo. Para este ejemplo, pondremos que el máximo para la coordenada en x es 1280 y el máximo para la coordenada y es 720. Por lo que **deberás validar** que, si te pasan valores negativos o mayores a los máximos permitidos (en constructores o setters), los pongas en 0.

TIP: valida en los setters y usa los setters en el constructor con parámetros (ya te doy el constructor con parámetros implementado, no lo modifiques)

1. Implementa el **constructor default** Este constructor inicializa todos los atributos en 0.
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 0 o mayor a los límites permitidos (1280 para posX y 720 para posY), le asignas por default un 0, 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 **igual(Posición&)**, devuelve un entero 1 si **“ésta”** Posicion es igual a la que **recibe como parámetro**. Decimos que son iguales, si coinciden en valores para los dos atributos. Regresa **0** en caso contrario.

CLASE PERSONAJE:

Representa los datos básicos de un Personaje (imagen a desplegar en una pantalla, en un videojuego), para este examen la clase está simplificada, solamente guarda el nombre del Personaje y la posición de este (coordenada donde se ubica su esquina superior izquierda de la imagen, sobre la pantalla).

1. Observa que el atributo **pos** es un **objeto de la clase Posicion**.
2. Codifica los **dos constructores (por defecto y con parámetros)**, así como **getters y setters** para todos los atributos. El nombre por defecto es **"Dummy"**.
3. El método **imprime()** imprime todos los datos del Personaje, incluyendo su posición. Utiliza el formato que tú quieras para esto (dale personalidad a tu **imprime**) y recuerda que tienes un método en la clase **Posicion** que imprime en el formato que tú diseñaste.
4. El método **avanzaHorz(int)** cambia la coordenada **x** del atributo **pos** (objeto **Posicion**) del Personaje, aumentando el valor que tiene, con el valor que recibe como parámetro. Por ejemplo, si el atributo **pos** de un Personaje ahorita tiene su coordenada en **X** con el valor de 5 y le aplicamos el **avanzaHorz** con el valor de 10 en el parámetro, el método deberá cambiar el valor del atributo **posX** del atributo **pos** del Personaje a 15 (5 coordenada en que estaba + 10 del avance que pide el método).

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

- En la función **main()** tienes declarada una **constante entera NUM**, la cual te servirá para el número de elementos de un arreglo.
- **(1 punto)** En la función **main()**, **declara un arreglo** de tipo **Personaje** 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 **pos** es un objeto de la clase **Posicion**.
- **(3 puntos)** **Completa la función imprimePersonajes** que sirve para desplegar a pantalla **TODOS** los datos de los objetos **Personaje** del arreglo. Usa la función correspondiente de **Personaje** que ya tienes codificada.
- **(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 **imprimePersonajes** para que se desplieguen a pantalla los datos de las **Personajes**.
- **(2 puntos)** Vamos ahora a "**avanzar a un Personaje**". Al **Personaje** que esté en la primera posición del arreglo aplícale el método para avanzar (que avance 15). **Imprime el Personaje** que avanzaste (misma posición en el arreglo), para ver si cambió adecuadamente de posición.
- **(2 puntos)** Escribe las instrucciones necesarias para determinar con un mensaje si el primer y último **Personaje** del arreglo están en la misma **Posicion** (usa el método adecuado de la clase **Posicion**).

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.