Universidad del Istmo de Guatemala
Facultad de Ingenieria
Ing. en Sistemas
Informatica II
Prof. Ernesto Rodriguez - erodriguez@unis.edu.gt

Laboratorio #5

Fecha de entrega: 28 de Febrero, 2019 - 11:59pm

Instrucciones: Resolver cada uno de los ejercicios siguiendo sus respectivas instrucciones. El trabajo debe ser entregado a traves de Github, en su repositorio del curso, colocado en una carpeta llamada "Laboratorio #5". Al menos que la pregunta indique diferente, todas las respuestas a preguntas escritas deben presentarse en un documento formato pdf, el cual haya sido generado mediante Latex. Este laboratorio debe ser elaborado en parejas.

Tarea #1 (25%)

En C++ es costumbre separar las declaraciones de clases y las implementaciones de las mismas. Adjunto a este laboratorio esta el archivo "Vector2d.hh", en el cual se declaran los *metodos* que debe implementar la clase "Vector2d". Utilize el archivo "Vector2d.cc" adjunto para implementar los metodos de la clase "Vector2d". El constructor ha sido implementado como ejemplo.

El metodo to_string de la clase vector debe retornar una *cadena* que sea una representacion del vector. Por ejemplo: "$\langle 2, 4.3 \rangle$".

Tarea #2 (25%)

Siguiendo el mismo patron de la Tarea #1, crear una clase llamada "Vehiculo" declarandola en un archivo de encabezados llamado "Vehiculo.hh" e implementandola en un archivo de codigo llamado "Vehiculo.cc".

El constructor de esta clase no debe recibir ningun parametro, sin embargo debe tener dos miembros *privados* llamados "Velocidad" y "Posición" de tipo "Vector2d". Ambos deben inicializarse con las coordenadas (0,0).

Declarar dos metodos:

- $getVelocidad$: `void` $\rightarrow$ `Vector2d`
- $getPosicion$: `void` $\rightarrow$ `Vector2d`

Estos metodos deben retornar la posición y velocidad respectivamente. Por ulitmo, declarar un metodo "to_string : `void` $\rightarrow$ `const std :: string`" que retorne una representación legible del vehiculo que muestre la velocidad y posición de dicho vehiculo. Recuerde que debe utilizar la directiva "`#include <string>`" para poder retornar valores de tipo *string*. Se recomienda que este metodo utilize el metodo "to_string" de los vectores del vehiculo para facilitar crear la cadena resultante.

Tarea #3 (20%)

Declarar en la clase "Vehiculo" el metodo *acelerar* : `const Vector2d&` → `const float` → `void`. Este metodo debe acelerar el vehiculo con la aceleración correspondiente al primer parametro siendo ese vector el cambio de velocidad por segundo. Dicha aceleración debe ser aplicada durante el tiempo indicado por el segundo parametro. Al finalizar la ejecución de este metodo, tanto la velocidad como la posición del vehiculo deben actualizarse tal que correspondan a la aceleración que tuvo el vehiculo.

Recuerde que la el archivo que declara la clase "Vehiculo" debe incluir la directiva #`include` "`Vector2d.hh`" para poder utilizar la clase "Vector2d".

Tarea #4 (20%)

Declarar en la clase "Vehiculo" el metodo "*avanzar* : `constfloat` → `void`". Este metodo debe utilizar la velocidad del vehiculo para moverlo durante la cantidad de tiempo que se dio como parametro. Luego de la ejecución de este metodo, la posición del vehiculo debe corresponder al movimiento durante el tiempo dado como primer paramtero.

Tarea #5 (10%)

Crear un archivo llamado "main.cc". En este arhivo debe declarar el metodo *main* a partir del cual se inica la ejecución del programa. Este metodo debe hacer lo siguiente:

1. Crear una *instancia* de un vehiculo
2. Acelerarlo en la dirección (3,1) durante 5 segundos
3. Mover el vehiculo a velocidad constante durante 10 segundos
4. Imprimir el estado actual del vehiculo utilizando su metodo "to_string"
5. Acelerar el vehiculo en la dirección (-7.2, 8) durante 4 segundos
6. Mover el vehiculo a velocidad constante durante 9 segundos
7. Imprimir el estado actual del vehiculo utilizando el metodo "to_string"

El archivo "main.cc" debe incluir los archivos "Vector2d.hh" y "Vehiculo.hh" para poder utilizar las clases "Vehiculo"y "Vector2d". El compilador (clang++) necesita que todos los archivos de codigo sean especificados para llevar a cabo la compilación por lo cual se debe compilar mediante el comando "clang++ Vehiculo.cc Vector.cc main.cc".