Universidad Panamericana
Campus México

Facultad de Ingeniería

Materia: LM0093 – Programación Orientada a Objetos
Proyecto: Segundo Parcial
Profesor: Mtro. Giancarlo Benítez V.

2do Proyecto Parcial

Mascota Virtual (Tipo – A v1.2)

Objetivos del proyecto:

- Aplicar los temas aprendidos durante el primer y segundo parcial en un proyecto de práctico
  - Combinar los conceptos: Herencias de clase, atributos y métodos estáticos, hilos y temporizadores, encapsulamiento y polimorfismo.
- Desarrollar un simulador de mascota virtual con las funciones comer, jugar, llevar al veterinario, pausar juego y empezar de nuevo.

Requerimientos:

- Crear una clase estática con todas las funciones de soporte:
  - Atributo estático que contenga la mascota virtual generada.
  - Menú de inicio con las opciones: continuar, juego nuevo, salir.
    - Continuar reiniciará los hilos de la mascota (no se puede continuar si no existe mascota)
    - Juego nuevo debe crear el huevo e incubarlo, luego iniciará todos los hilos necesarios.
    - Cargar juego: leerá los datos de la mascota desde un archivo y automáticamente dará inicio a "continuar" el juego. En caso de que ya exista una mascota en juego, lanzar advertencia que perderá los cambios no guardados y si desea continuar.
    - Guardar juego: escribe todos los datos necesarios, para el correcto funcionamiento del juego, en un archivo.
    - Salir terminará la ejecución del programa, así como también todos los hilos creados de la mascota.
  - Nueva mascota: crea un nuevo huevo de una clase aleatoria
  - Menú de juego: muestra los atributos de la mascota (vida, hambre y diversión), permite escoger entre las opciones (alimentar, jugar, veterinario y regresar al menú)
    - La opción regresar al menú principal debe detener todos los hilos en ejecución
- Crear la clase padre Mascota y clases hijos (al menos 3) con modificadores de vida, hambre y aburrimiento diferentes.
  - Declarar los atributos públicos, privados y protegidos correspondientes.
  - Al crear una nueva mascota se determina de manera aleatoria el tipo de huevo creado.
  - A uno le dará más hambre más rápido, otro se aburrirá más rápido y el último perderá vida más rápido.
    - Sugerencia: pueden utilizar un ponderador en la fórmula para reducir/aumentar según el tipo de criatura.
- Debe contener los atributos: nombre, salud, diversión, hambre, pérdida de diversión y ganancia de hambre, diccionario de hilos y tiempo de incubación.
- Funciones:
  - Incubar: se debe generar un tiempo de incubación aleatorio entre 5 y 10 segundos, luego eclosionar, solicitar nombre de mascota e iniciar los hilos respectivos para generar hambre y aburrimiento.
  - Iniciar hilos: inicia los hilos respectivos para generar hambre y aburrimiento.

- Detener hilos: detiene todos los hilos del objeto.
- Generar hambre: es un hilo recursivo que debe aumentar el contador de hambre de la mascota y calcular una pérdida de vida porcentual a la cantidad de hambre
  - Se sugiere (vida – ¼ hambre) aunque pueden redefinir esta función.
- Generar aburrimiento: es un hilo recursivo que debe disminuir la cantidad de diversión de la mascota y una pérdida de vida porcentual al aburrimiento.
  - Se sugiere (⅛ (100-diversión)) aunque pueden redefinir esta función.
- Llevar al veterinario: aumentará la mitad de la vida de la mascota, pero reducirá la diversión en un tercio.
- Alimentar: disminuirá el hambre en 5 unidades
- Jugar: aumentará la diversión en 5 unidades.

Entregables:

- Todos los archivos *.py deberán ser entregados en una carpeta comprimida .zip con el nombre:
  - <nombre>_<apellido>_MascotaP2.zip
- No se aceptan entregas fuera del plazo establecido.
- Se podrá hacer una revisión y resolución de dudas el día jueves 31 de marzo durante la clase
  - Adicionalmente pueden agendar una asesoría en la liga: https://calendly.com/gbenitez-talent/30min
- **La entrega será en Moodle el día lunes 4 de abril**
- **El examen parcial será el martes 5 de abril**