

Práctica de laboratorio #3 – lógica y representación 1

Ingeniería de sistemas – UdeA 2023

La presente práctica tiene como finalidad evaluar el entendimiento adquirido por los estudiantes sobre **Programación orientada a objetos**, sus fundamentos y cómo desarrollarlo en programación.

El estudiante deberá desarrollar un sistema que permita dar solución al problema que se planteará a continuación, en el lenguaje de programación C#.

El estudiante deberá saber explicar sin problema el funcionamiento del sistema realizado y responder a las preguntas que el profesor y/o la monitora le realicen, demostrando así que comprende el tema.

Contexto

La programación orientada a objetos nos permite realizar interpretaciones de objetos o eventos del mundo real como entidades de programación que abstraen sus características y comportamientos fundamentales. Para este desarrollo vamos a simular el comportamiento de un juego de amigo secreto por medio de la programación orientada a objetos.

Utilizando interfaz gráfica, programa principal deberá poder simular un juego utilizando un objeto de la clase AmigoSecreto. Para realizar la simulación el programador deberá usar métodos de la clase AmigoSecreto y Jugador, con el fin de que un usuario pueda asignar la cantidad de jugadores, fecha inicio, fecha, número de endulzadas, frecuencia de endulzadas en días, valor de la endulzada, valor del regalo, asignar los jugadores con sus gustos, repartir los amigos secretos, imprimir la información del juego, imprimir los gustos de cada jugador y dada una fecha ingresada por el usuario calcular en cuantos días es la próxima endulzada.

Se debe crear una clase Jugador que tenga las características de: Nombre, correo, endulzada ideal, regalo ideal, amigo secreto. Los métodos de esta clase serán los setters y getters correspondientes.

Se debe crear una clase AmigoSecreto que tenga las características de: Cantidad de jugadores, fecha inicio, fecha fin (pueden usar el nombre de su preferencia como, por ejemplo: descubrimiento), número de endulzadas, frecuencia de endulzadas en días, valor de la endulzada, valor del regalo y finalmente un vector de objetos de la clase Jugador mencionada anteriormente. Lo métodos de esta clase serán los setters y getters correspondientes, además, se necesita:

- Un método para asignar los jugadores usando la variable “cantidad de jugadores”.
- Un método para asignar los amigos secretos y simular para cada jugador un mensaje de correo electrónico informando el amigo secreto (no es válido que un jugador sea su propio amigo secreto)
- Un método para imprimir la información del juego (fecha inicio, fecha fin, número de endulzadas, frecuencia de endulzadas en días, valor de la endulzada, valor del regalo).
- Un método para imprimir los gustos de cada jugador.
- Un método para calcular en cuantos días es la próxima endulzada ingresando una fecha como parámetro.

Todo lo relacionado con fechas debe ser de tipo `DateTime`, una clase de C#, aquí encuentran más información. <https://www.c-sharpcorner.com/article/datetime-in-c-sharp/>

Finalmente, deberá subir el sistema creado en C# a un repositorio público de GitHub, realizar un manual de usuario (Documento de guía para el usuario final sobre cómo usar adecuadamente el

aplicativo) y realizar la documentación de programador por **cada clase y método** implementado en la solución.

Evaluación

Usa GitHub como repositorio del código.	20%
Usa correctamente el estilo de escritura Camel case, además, nombra claramente las variables y métodos de acuerdo con el propósito de cada uno.	10%
La solución cumple con la simulación completa del juego.	60%
Documentación de la solución y manual de usuario	10%

BONUS: Todo lo que se haga adicional será tomado en cuenta.

Ante cualquier duda se puede comunicar por los siguientes medios:

- **E-mail:** sofia.vanegasc@udea.edu.co (preferiblemente por el chat del correo)
- **WhatsApp:** 3156835552

Recuerde que la fecha límite de entrega es el jueves 19 de octubre de 2023 a las 23:59, debe ser enviado al correo electrónico mencionado antes con el asunto **[LAB3-LR1]**, debe ser enviado mediante un enlace de Github y el repositorio debe ser público.

¡Muchos Éxitos!

Trabajos con copia tendrán consecuencias >:)