Preguntas Teóricas (Parcial 1 de POO),

Juan Francesco García Vargas(código: 8963676)

ENCAPSULAMIENTO:

Realmente no es necesario que sea protected, aunque puede declararse el método protegido, ya bien sea en público o protegido, al poner la palabra virtual y la definición de sobreescritura(override) el método se sobrescribe y es polimorfismo, esto es más claro ya que en la clase padre "Juego", se declara como abstracto. Así que al ser un método y tener el virtual ya se puede sobrescribir en las clases hijas.

CONTENEDORAS:

Por supuesto que estoy de acuerdo con usar un <unordered_map> para organizar los jugadores en el casino. Pues teniendo en cuenta que no existe un orden importante para organizar los jugadores del casino, y que estos jugadores van a ser verificados e identificados a partir de un id, entonces es el perfecto caso para usar un <unordered_map> . La complejidad computacional del mapa no ordenado es de O(1), por lo que es más optimo usar este en caso de no necesitar orden, así haciendo un programa más optimo y/o eficaz. Las demás opciones necesarias como, mostrar los datos del jugador, recargar sus Gonzos, verificar que el jugador exista y eliminar un jugador se muestran buscando la llave en el <unordered_map>.

CLASES ABSTRACTAS:

Explique qué cosas del código fuente indican que la clase es una clase abstracta

La clase juego debe ser una clase abstracta porque así se muestra en el enunciado indirectamente. Como calculo las ganancias en Gonzos de un jugador si antes no sé qué juego está jugando? Si todos los juegos tuvieran las mismas reglas y ganancias entonces no sería abstracto, sin embargo, a partir del juego es que se puede calcular las ganancias o pérdidas de un jugador, porque lo que clase juego debe ser abstracta. Esto dentro del código fuente se puede verificar con la igualación del método a cero, de la siguiente manera, "virtual float calcularResultado(float gonzosApostar) = 0;". Al ver esto podemos decir inmediatamente que estamos ante una clase abstracta.

¿Qué implicaciones tiene esa decisión de diseño para este programa? Explique

Realmente los cambios son simplemente de sobreescritura. Ya que al "Juego" ser clase abstracta pues primero se debe instanciar un juego (cualquiera de las clases hijas de juego, por ejemplo, el Mayor de 13) y luego acceder a su función virtual calcularResultado(float). Dentro del diseño, como se mencionó en un punto anterior, puede ponerse como un método público o como protegido, siempre y cuando sobrescriba y tenga polimorfismo para las clases hijas.

SOBRECARGA-ESCRITURA:

Fácilmente se puede identificar (en el código fuente) por los métodos con la palabra reservada "override", los cuales los poseen tanto la clase "Mayor13", como "DosColores" en el método sobrescrito, calcularResultado(float gonzosApostar). Dentro del diseño, pues se identificará porque dentro de la clase padre "Juego" y de sus dos clases hijas existe el mismo método con los mismos parámetros.