



Profesores:

- Francisco Arroyo
- Edgar Casasola
- Braulio Solano
- Javier Vásquez
- Maureen Murillo (coordinadora)

**Examen #1 de Cátedra
II-2018**

Observaciones

1. Cuenta con 3 horas para entregar su solución del examen.
2. El examen es individual. Es prohibido utilizar herramientas digitales para intercambio de documentos.
3. Guarde su celular. Si lo utiliza para lo que sea se considerará fraude.
4. Puede utilizar cualquier material propio (prácticas, tareas, libros, apuntes) y material de referencia en Internet.
5. Como nombre de su proyecto indique su carne junto con ex1, ej: B82345ex1.
6. En el comentario principal de cada clase, como autor indique su número de carné y su nombre.
7. En cada clase, incluya comentarios con el número de pregunta que está respondiendo (marque el inicio y el fin de cada respuesta). Ej:
//inicio respuesta a pregunta #1 15%.
8. Cada 20 minutos suba su solución en un archivo comprimido que contenga el código fuente al sitio de entrega del examen de su grupo.
9. En caso de fallo de energía eléctrica deben continuar su examen en papel. Se calificará la versión subida hasta ese momento a la página del grupo, así como la documentación que aporte en su cuaderno de examen
10. La solución del exámen debe subirse dentro del tiempo asignado, a la página usada en su grupo. Al finalizar su examen y por motivos de seguridad guarde en su cuenta de correo, archivo drive o memoria USB una copia de la solución entregada.
11. Firme la hoja de asistencia al finalizar el examen.

Enunciado

Construya un programa que implemente el juego de cartas “Veinticinco”, en el cual el usuario podrá jugar contra la computadora. Una partida del juego consiste en repartir una mano de 5 cartas para cada jugador (usuario y computadora). La partida la gana el jugador que tenga la mano más cercana al valor 25. Note que se puede aproximar a 25 desde cantidades inferiores o superiores. Por ejemplo, si la mano del usuario suma 20 y la de la computadora suma 28, la computadora gana la partida.

Su programa debe seguir las siguientes especificaciones, pudiendo añadir más clases o los métodos

que considere necesarios:

1. Programe una clase llamada **"Carta"** que represente una carta del naipes. Indique claramente los atributos que debe tener esta clase. La carta tiene un valor (A,2,3,4,5,6,7,8,9,10,J,Q,K) y un palo (treboles ♣, corazones ♥, diamantes ♦, espadas ♠). 4%
2. La clase Carta debe tener un método **constructor** que recibe como parámetros los valores necesarios para inicializar sus atributos. 3%
3. Además, debe tener un método **"String toString()"** que retorna una hilera de caracteres con una representación de la carta, por ejemplo: "As de corazones". 3%
4. Esta clase debe tener un método **"int getValor()"** que retorna el valor de la carta: 5%
 - a. De 2 a 10 para aquellas cartas con número.
 - b. 10 para aquellas con "J", "Q" y "K".
 - c. 11 para aquellas con "A".
5. Implemente una clase denominada **"Baraja"** que, usando arreglos (vectores), represente una baraja de naipes o cartas (52 cartas, agrupadas en los 4 palos anteriormente indicados). 5%
6. Esta clase debe tener un constructor **"Baraja()"** que permita crear la baraja (conjunto de todas las cartas que la conforman y que no se repiten). 10%
7. Para esta clase debe construir un método **"baraje()"** que revuelva la baraja de manera que las cartas que se van repartiendo queden al azar. 15%
8. Además, debe existir un método **"Carta demeCarta()"** que permita tomar la siguiente carta de esta baraja, es decir que retorna la carta la baraja y la elimina de las cartas disponibles para ser jugadas. 10%
9. Construya una clase **"Mano"** que contenga cinco cartas de la baraja almacenadas en un arreglo, así como los métodos para asignar (set) y recuperar (get) el valor de la misma 10%
10. En la clase Mano implemente un método **"boolean gana(Mano segundaMano)"** que devuelve "true" si la primera mano le gana a la segunda (pasada como parámetro a este método). En caso de empate, gana la primera mano. Puede construir un método adicional **"getValor()"** que devuelva el valor numérico de la mano y le facilite la comparación de las manos. 15%
11. Construya la(s) clase(s), que permita(n) ejecutar una serie de hasta 5 juegos entre la computadora y una persona, de manera que cada uno reciba una mano y el programa determine el ganador de cada juego y de la serie. 20%

Principales aspectos a evaluar dentro del puntaje de cada pregunta:

1. Cumplimiento de buenas prácticas de programación.
2. Separación de responsabilidades de los objetos.
3. Correcta especificación de clases.
4. Uso correcto de constructores.
5. Uso correcto de la programación orientada a objetos.
6. Uso de arreglos de objetos.
7. Uso correcto de estructuras de control.

8. Entrada/salida.