

EXAMEN 1ª AVALUACIÓ -PRÀCTICA

MODUL - PROGRAMACIÓ

CURS: 2023/2024

Autors: Nombre professors: Guillermo Garrido Portes

Jose Cantó Alonso

Salvador Rue Orquin

Correus electrònics : g.garridoportes@edu.gva.es

j.cantoalonso@edu.gva.es

s.rueorquin@edu.gva.es

Llicència



[CC BY-NC-SA 3.0 ES](https://creativecommons.org/licenses/by-nc-sa/3.0/es/) Reconeixement – No Comercial – Compartir Igual (by-nc-sa)

No es permet un ús comercial de l'obra original ni de les possibles obres derivades, la distribució de les quals s'ha de fer amb una llicència igual a la que regula l'obra original. Aquesta és una obra derivada de l'obra original de Carlos Cacho i Raquel Torres.

1. INTRODUCCIÓ

Anem a crear un projecte per a jugar al Blackjack. Com que el joc es prou complex s'ha simplificat pel que per a aquesta prova sols es demana part del joc.

Per fer-ho s'han de seguir les instruccions exactament.

2. NOMENCLATURA

- Les funcions que es creen hauran de seguir la nomenclatura que s'indica en l'exercici.
- S'ha de ficar el nom i cognom a l'etiqueta @author
- Les variacions en nomenclatura i estructura es penalitzaran.
- El arxiu final s'entregarà en la tasca d'aules que indique el professor.

3. ENTORN DE PROGRAMACIÓ

- S'utilitzarà el IDE de programació online codeboard.
- L'enllaç d'accés al projecte és <https://codeboard.io/projects/422137>
- Es pot compilar i fer-hi submit tots els viatges que siguen necessaris
- En finalitzar es copiarà el codi i es pujarà al quadre de text habilitat en Aules.

4. PROGRAMA

- El programa ha de importar de la llibreria de java **util** les classes **ArrayList**, **HashMap** i **Scanner**.
- El programa compta amb dos variables globals, **clients** que serà un HashMap i baralla que tindrà la següent estructura: **baralla = {"1", "2", "3", "4", "5", "6", "7", "8", "9", "sota", "cavall", "rei"};** Per a evitar que el jugador puga contar cartes, la baralla tindrà un nombre infinit de cartes de cada tipus, pel que no cal contemplar el numero de cartes de cada tipus que hi ha en joc.
- La classe HashMap compta amb la funció **containsKey** que rebrà la clau del mapa com a paràmetre i torna verdader si existeix o fals en cas contrari.
- La classe Scanner sempre es crearà utilitzant el paràmetre de **System.in**.

Pregunta 0: Afegir Client (2 punts)

La funció **main()** afegirà un jugador i començarà la partida amb **jugarBlackjack()**.

Pregunta 1: Afegir Client (3 punts)

Implementa una funció anomenada **afegirClient()** que reba el nom del client i el seu saldo, i l'afija al HashMap clients.

Pregunta 2: Mostrar Client (5 punts)

Desenvolupa una funció denominada **mostrarClient()** que, donat el nom d'un client, imprimisca en la consola el seu nom i saldo. Si el client no es troba en el sistema, haurà de mostrar un missatge indicant que el client no ha sigut trobat.

Exemple, intentarem buscar amb la funció a Juan (introduït prèviament amb saldo 500) i a Carlos, mostraria:

Client: Juan, Saldo: 500.0 euros.

Client amb nom Carlos no trobat.



Pregunta 3: Modificar Saldo (10 punts)

Crea la funció **modificarSaldo()**, la qual ha de rebre el nom del client i una quantitat. La funció haurà de modificar el saldo del client corresponent en el HashMap clients, afegint la quantitat indicada. Assegura't de manejar adequadament el cas en què el client no existisca.

Exemple, si basant-nos en els exemples anteriors intentarem modificar amb la funció el saldo de Juan i a Carlos llevant-li 20€ a cadascun, mostraria:

Client: Juan, Saldo: 480.0 euros.

Client amb nom Carlos no trobat.

Pregunta 4: Obtindre Carta Aleatòria (10 punts)

Defineix una funció anomenada **obtindreCartaAleatoria()** que retorna un número aleatori entre 0 i 11. Este número representarà el valor d'una carta en el joc de Blackjack.

Pregunta 5: Prendre Carta (10 punts)

Crea la funció **pendreCarta()**, que rep un ArrayList representant la mà d'un jugador i afeg una carta aleatòria a eixa mà utilitzant la funció **obtindreCartaAleatoria()**.

Pregunta 6: Calcular Valor Cartes (15 punts)

Desenvolupa la funció **calcularValorCartes()**, que rep un ArrayList amb les cartes d'un jugador i retorna el valor total d'eixes cartes. Utilitza el menor nombre de estructures condicionals per a complir amb les regles de les cartes següents:

- La sota, cavall i rei tenen un valor de 10.
- El as (1) te el valor de 11.
- La resta de números tenen el valor que indica en la mateixa carta.

Exemple, si un jugador tingues les cartes *[7, rei, 1]*, tornaria un valor de *[7, 10, 11] = 28*.

Pregunta 7: Mostrar Resultats Blackjack (15 punts)

Completa la funció **mostrarResultatsBlackjack()**, que rep les cartes del jugador i del crupier, calcula i mostra en la consola les cartes i els resultats finals de la partida, incloent-hi el guanyador segons les regles del Blackjack.

Exemple, si a la funció li arriben els dos ArrayList amb els següents valors *[7, rei, 1]* i *[8, 2, 9]*, esta mostrarà:

Cartes del jugador: [7, rei, 1]

Cartes del crupier: [8, 2, 9]

*Valor de las cartas del jugador: 28 Valor
de las cartas del crupier: 19 ¡Crupier
guanya!.*

Pregunta 8: Jugar Blackjack (30 punts)

Implementa la funció **jugarBlackjack()**, seguint les següents indicacions:

- Crea 2 ArrayList per a emmagatzemar les cartes del jugador i del crupier.
- El jugador i crupier prenen una carta aleatòria al inici.
- La funció ha de sol·licitar al jugador si desitja prendre una altra carta o no.
- Si el jugador decideix que si,
 - Pren una carta aleatòria i es mostra el seu valor.
 - Es verifica si la ma del jugador supera 21.
 - Si és així, es mostra el resultat.
 - En cas contrari, el crupier prendrà una carta.
 - Es verifica si la ma del crupier supera 21.
- Si és així, es mostra el resultat.
- Si el jugador decideix que no, el crupier prendrà cartes fins que la seua puntuació siga igual o superior a 17.
- Finalment, es mostra el resultat.