CFGS: Desenvolupament d'Aplicacions Multiplataforma
M03: Programació
Cognoms: Batalla
Nom: Edim

Qualificació:

Data: 20/01/2025

P2: Estructura de la programació avançada - Arrays.

Activitat 1 - Gestió avançada d'un banc amb arrays.

Després de definir el **metode main**, importar i inicialitzar l'objecte **scanner** definim la variable **double saldo = 0;** que servirà per guardar el saldo actual del compte.
També creem un array per registrar els moviments, amb capacitat de 100 moviments:
double[] moviments = new double [100];
Definim una altre variable tipus integer per comptar els moviments, **int numMoviments = 0;**

Ara definim un bucle **while** que s'executarà de manera indefinida mentres la condició **(true)** es mantingui. El bucle executa la impressió de les línies de text en les que informem a l'usuari de les diferents opcions que té en el menu del compte bancari:

```
System.out.println("\n--- Compte Bancari ---");
System.out.println("\n1.Consulta saldo");
System.out.println("2.Ingressar");
System.out.println("3.Retirar");
System.out.println("4.Consultar històric");
System.out.println("5.Consultar major ingrés");
System.out.println("6.Consultar major retirada");
System.out.println("7.Consultar totals");
System.out.println("8.Determinar Perfil");
System.out.println("9.Sortir");
System.out.println("\nSelecciona una opció:");
```

Després de mostrar el menú es demana a l'usuari que seleccioni una opció mitjançant el número corresponent amb l'opció desitjada. La selecció de l'usuari es guarda amb la variable utilitzant l'objecte scanner: **int opcio = scanner.nextInt();**

Aquí s'inicia una estructura de control **switch (opcio)** basada en el valor de la variable **int opcio = scanner.nextInt();**.
Aquesta estructura consta de les 9 variables que trobem en el menu del compte bancari i una de predeterminada per si l'usuari selecciona una opció no contemplada en el codi per error.

Case 1
Si l'usuari prem l'opció 1 s'activarà el **case 1:**
System.out.println("\nEl teu saldo és: "+saldo+"€");
break;
En el que mitjançant **System.out.println** s'imprimeix el valor de saldo. La instrucció **break** assegura que el programa surti del switch un cop executat aquest cas.

Case 2
Quan l'usuari prem l'opció 2 s'activarà el **case 2:** que serveix per ingressar diners.
Primerement es demana a l'usuari quina quantitat vol ingressar mitjançant:
System.out.println("\nQuina quantitat vols ingressar?");
I utilitzant la funció de l'scanner es guarda la quantitat a la següent variable:
double ingres = scanner.nextDouble();

Amb el condicional **if (ingres < 0)** comprovem que l'ingrés sigui més alt que zero, osigui que si l'ingrés és zero o un número negatiu ens informa que la quantitat es errónea amb el següent missatge:
System.out.println("\nL'import ingressat ha de ser positiu. Torna-ho a intentar.");

A l'enunciat de l'exercici ens demana que si el saldo supera els 10000€ s'apliqui una comissió del 2%. Jo ho he aplicat de la següent manera, he considerat que aquesta comissió s'hauria d'aplicar quan fem un ingrés de més de 10.000€ al compte bancari.
LLavors en el mateix case dos utilitzem el condicional **} else if (ingres > 10000) {** per que s'activi la part del codi d'aquest condicional el primer que s'ha de cumplir es que no s'hagi activat el primer condicional del **case 2** que es: **if (ingres < 0)**.
Això vol dir que si l'ingrés és positiu i es major a 10000€ automàticament s'activaran les següents línies de codi que formen part d'aquest condicional.

Es declara la variable **double comisio = ingres*0.02;** que consisteix a aplicar un 2% de l'import ingressat.
Després s'executa la operació **ingres -= comisio;** en la que a la quantitat que s'està ingressant se l'hi resta la quantitat que la comissió que hem calculat amb la variable anterior.
A continuació sumem l'import final de l'ingrés al saldo actual del compte
saldo += ingres;

Mitjançant les línies de codi de a continuació s'imprimeix la informació del ingres, la comissió i el saldo actualitzat per que l'usuari estigui informat:
System.out.println("\nAl superar l'ingrés la quantitat de 10.000€ s'aplica una comisió del 2% que en aquest cas és de "+ comisio +"€");
System.out.println("S'han ingressat " + ingres + " €");
System.out.println("El teu saldo és "+saldo+"€");
Per acabar aquest condicional es registra l'import de l'ingrés en l'array moviments:
moviments[numMoviments++] = ingres;

Si quan es fa un ingrés cap dels dos condicionals anteriors s'activa ja que l'ingrés es positiu i menor a 10000€ s'activarà aquest últim condicional **} else {**
Aquesta línia suma l'import de l'ingrés al saldo: **saldo += ingres;**
S'informa al usuari de les operacions amb la impressió de les següents líneas:
System.out.println("\nS'han ingressat " + ingres + " €");
System.out.println("El saldo actual és: " + saldo + " €");
I com en el condicional anterior registrem l'import de l'ingrés en l'array moviments:
moviments[numMoviments++] = ingres;
Amb la instrucció **break;** sortim de l'estructura switch.

Case 3
Aquest cas tè la funció de Retirar diners. Primer es demana a l'usuari que introdueixi la quantitat que vol retirar i es guarda a la variable retirar mitjançant el scanner.
System.out.println("\nQuina quantitat vols retirar?");
double retirar = scanner.nextDouble();

A partir d'aquí, es gestionen diversos casos amb condicions **if** i **else if** per assegurar que la retirada sigui vàlida:

Si el saldo és menor o igual a 0, el programa mostra un missatge informant que no es poden retirar diners perquè no hi ha saldo disponible.
if (saldo <= 0) {
System.out.println("\nNo es poden retirar diners perquè el saldo és insuficient.");

Si l'usuari intenta retirar una quantitat menor o igual a 0, es mostra un missatge indicant que l'import ha de ser positiu.
} else if (retirar <= 0) {
System.out.println("\nL'import a retirar ha de ser positiu. Torna-ho a intentar.");

Si l'import a retirar és superior al saldo disponible, es notifica que no es pot completar la retirada perquè excedeix els diners disponibles al compte.
} else if (retirar > saldo) {
System.out.println("\nNo es pot retirar aquesta quantitat perquè excedeix el saldo disponible.");

Si cap de les condicions anteriors es compleix, es realitza la retirada. Es resta l'import de la variable saldo amb saldo **-= retirar**, es mostra el nou saldo actual i es registra la retirada com un moviment negatiu a l'array moviments, utilitzant **moviments[numMoviments++] = -retirar.**

Finalment, la instrucció **break;** assegura que el programa surti del switch un cop acabada l'operació.

Case 4
Aquest cas correspon a l'opció Consultar l'històric de moviments. Quan l'usuari selecciona aquesta opció, el programa mostra tots els moviments que es troben guardats a **l'array moviments.**

Amb un bucle **for** que recorre les posicions de l'array des de **0** fins a **numMoviments,** que indica quants moviments s'han registrat fins ara. Per a cada moviment, es mostra el número del moviment (començant per 1) i l'import corresponent.
System.out.println("\nHistòric de moviments:");
for (int i = 0; i < numMoviments; i++) {
System.out.println("\n Moviment" + (i+1) + "."+ moviments[i] +"€");
}

Acabem el case un altre cop amb la instrucció **break**; per sortir del cas del switch.

Case 5
Amb aquest cas consultem el major ingrés. El programa busca l'ingrés més alt registrat a **l'array moviments** i el mostra a l'usuari.

Primer, es declara la variable maxPositiu i se li assigna el valor inicial de 0.
double maxPositiu = 0;
Aquesta variable s'utilitzarà per guardar l'import positiu més gran trobat.
A continuació, es recorre l'array moviments amb un bucle for.
for (int i = 0; i < numMoviments; i++) {
Amb la següent condició només s'utilitzaran els valors positius i que el moviment sigui amb el major valor del emmagatzemats
if (moviments[i] > 0 && moviments[i] > maxPositiu) {
I si el valor actual és més gran que **maxPositiu**, s'actualitza amb aquest import.
maxPositiu = moviments[i];

Acabem el case un altre cop amb la instrucció **break**; per sortir del cas del switch.

Case 6
Aquest cas correspon a l'opció consultar la retirada més gran.

Primer, declarem la variable maxNegatiu, inicialitzada a 0, que s'utilitzarà per guardar el moviment negatiu més gran trobat.
double maxNegatiu = 0;
A continuació, es recorre l'array moviments amb un bucle for.
for (int i = 0; i < numMoviments; i++) {
Amb el bucle es revisen només els valors negatius mitjançant i que sigui el negatiu mes alt.
if (moviments[i] < 0 && moviments[i] < maxNegatiu) {
Si el valor actual és més negatiu que el valor de maxNegatiu, s'actualitza maxNegatiu amb aquest valor.
maxNegatiu = moviments[i];

Després del bucle, es comprova amb un condicional if si maxNegatiu continua sent 0.
if (maxNegatiu == 0) {
Si és així, significa que no s'han registrat retirades, i es mostra un missatge informant l'usuari.
System.out.println("\nNo hi ha retirades registrades.");
Si s'ha trobat alguna retirada, mitjançant el condicional else es mostra la retirada més gran amb un missatge que indica el seu valor.
} else {
System.out.println("\nEl moviment negatiu més gran és: " + maxNegatiu + "

Com amb la resta de casos utilitzem la instrucció **break;** per sortir del cas.

Case 7
Aquest cas correspon a l'opció consultar els totals d'ingressos i retirades.

Primer, declarem dues variables, sumaIngressos i sumaRetirades, inicialitzades a 0, que serviran per acumular els valors dels ingressos i les retirades respectivament.
double sumaIngressos = 0;
double sumaRetirades = 0;

A continuació, s'utilitza un bucle for per recorrer tots els elements de l'array moviments.
for (int i = 0; i < numMoviments; i++) {
Si el moviment actual és positiu, s'afegeix a sumaIngressos.
if (moviments[i] > 0) {
sumaIngressos += moviments[i];
Si en canvi és negatiu s'afegeix a sumaRetirades.
if (moviments[i] > 0) {
sumaIngressos += moviments[i];

Un cop completat el bucle, el programa mostra la suma total dels ingressos i la suma total de les retirades.
System.out.println("\nSuma total d'ingressos: " + sumaIngressos + " €");
System.out.println("\nSuma total de retirades: " + sumaRetirades + " €");

Com amb la resta de casos utilitzem la instrucció **break;** per sortir del cas.

Case 8
Aquest cas correspon a l'opció determinar el perfil del compte.

Primer, es declaren dues variables, totalIngressos i totalRetirades, inicialitzades a 0.
double totalIngressos = 0;
double totalRetirades = 0;
Aquestes variables acumularan respectivament la suma de tots els ingressos i la suma de totes les retirades.

Mitjançant un bucle for, recorrem l'array moviments per processar cada moviment:
for (int i = 0; i < numMoviments; i++) {
Si el moviment actual (moviments[i]) és positiu, s'afegeix a totalIngressos.
if (moviments[i] > 0) {
totalIngressos += moviments[i];
Si el moviment és negatiu, el seu valor absolut s'afegeix a totalRetirades.
} else {
totalRetirades += -moviments[i];

Un cop finalitzat el bucle, comparem els valors totals:
Si els ingressos totals són més grans que les retirades, el compte es classifica com a estalviador.
if (totalIngressos > totalRetirades) {
System.out.println("\nEl perfil del compte és: Estalviador");
Si les retirades totals superen els ingressos, el compte es classifica com a gastador.
} else if (totalRetirades > totalIngressos) {
System.out.println("\nEl perfil del compte és: Gastador");
Si ingressos i retirades són iguals, el compte es considera equilibrat.
} else {
System.out.println("\nEl perfil del compte és equilibrat.");

Case 9
El cas 9 serveix per sortir del programa.

Es mostra un missatge informant que el programa s'està tancant amb **System.out.println("\nSortint del programa...").**

Es tanca l'objecte scanner amb **scanner.close()**

Finalment, s'utilitza **return;** per sortir del mètode main i acabar l'execució del programa.

Case Default:
Si l'usuari prem una opció no contemplada en el sistema, osigui que prem una opció erronea s'informa l'usuari que la opció és incorrecte i que ho torni a provar.
System.out.println("\nOpció no vàlida, torna a provar.");

Tornem a utilitzar la instrucció **break;** per sortir del cas.

Activitat 2 - Gestió avançada d'una biblioteca amb arrays.

Després de definir el **metode main**, importar i inicialitzar l'objecte **scanner** definim un array de cadenes de text per emmagatzemar els títols dels llibres disponibles. L'array pot contenir fins a 100 títols diferents.
String[] titols = new String[100];
Un array de booleans que indica si cada llibre està prestat o no.
boolean[] prestat = new boolean[100];
Un array per emmagatzemar el nom dels usuaris que han prestat els llibres.
String[] usuaris = new String[100];
Un array de que guarda el nombre de vegades que s'ha prestat cada llibre.
int[] numPrestecs = new int[100];
I una variable que actua com a comptador del nombre total de llibres registrats actualment al sistema.
int numLlibres = 0;

Ara definim un bucle **while** que s'executarà de manera indefinida mentres la condició **(true)** es mantingui. El bucle executa la impressió de les línies de text en les que informem a l'usuari de les diferents opcions que té en el menu

```
while (true) {
System.out.println("\n--- Gestió Biblioteca ---");
System.out.println("\n1. Consultar estat dels llibres");
System.out.println("2. Afegir un nou llibre a la bibloteca");
System.out.println("3. Realitzar un préstec");
System.out.println("4. Gestionar retorn");
System.out.println("5. Consultar l'històric de prestecs");
System.out.println("6. Llibres més solicitats (Top 5)");
System.out.println("7. Sortir");
System.out.print("\nSelecciona una opció: ");
```

Mitjançant l'objecte scanner registrem la opció desitjada a la variable opcio.
int opcio = scanner.nextInt();

Iniciem la estructura de control **switch (opcio)**

Case 1
Consultar l'estat dels llibres, utilitzem un bucle for que recorre els llibres fins al nombre actual de llibres registrats.
for (int i = 0; i < numLlibres; i++) {

Definim la **variable estat** per emmagatzemar l'estat actual d'un llibre. A través d'una estructura **condicional if-else** es comprova si el llibre a la posició i de l'array està prestat. Si la condició és certa a estat se li assigna la cadena "Prestat a: " seguida del nom de l'usuari que té el llibre extret de **l'array usuaris[i]**. Si la condició és falsa significa que el llibre està disponible.
String estat;
if (prestat[i]) {
estat = "Prestat a: " + usuaris[i];
} else {
estat = "Disponible";

Case 2
Amb quest afegim un nou llibre a la biblioteca.
El programa comprova primer si encara hi ha espai disponible per afegir més llibres, ja que el límit està fixat en 100 llibres.
if (numLlibres < 100) {

Si encara es poden afegir llibres, es mostra un missatge perquè l'usuari introdueixi el títol del llibre. El títol es llegeix amb l'objecte scanner i es guarda a la variable titol.
System.out.print("\nIntrodueix el títol del llibre: ");
scanner.nextLine();
String titol = scanner.nextLine();

Aquest títol s'assigna a l'array titols a la posició indicada per numLlibres.
titols[numLlibres] = titol;

I s'inicialitzen els altres arrays associats a aquest llibre:
prestat[numLlibres] = false; es defineix com false ja que el llibre no està prestat.
usuaris[numLlibres] = ""; es deixa buit perquè encara no té cap usuari assignat.
numPrestecs[numLlibres] = 0; es fixa en 0, ja que el llibre encara no ha estat prestat.

Després d'inicialitzar aquesta informació, el comptador numLlibres s'incrementa per registrar el nou llibre al sistema, i es mostra un missatge confirmant que s'ha afegit correctament.
numLlibres++;
System.out.println("\nLlibre afegit a la biblioteca.");

Si no hi ha espai disponible es mostra un missatge indicant que la biblioteca està plena i no es poden afegir més llibres.
} else {
System.out.println("\nNo es poden afegir més llibres. Biblioteca plena.");

Case 3
Aquest bloc gestiona el préstec d'un llibre de la biblioteca. El procés comença demanant a l'usuari que introdueixi el títol del llibre que vol prestar, que es llegeix amb el scanner i s'emmagatzema a la variable llibrePrestec.
System.out.print("\nIntrodueix el títol del llibre per realitzar el préstec: ");
scanner.nextLine();
String llibrePrestec = scanner.nextLine();

A continuació, es declara la variable llibreTrobat, inicialitzada com false, per fer un seguiment de si el llibre està disponible per al préstec.
boolean llibreTrobat = false;

Es recorre l'array titols amb un bucle for que busca si el títol introduït existeix en la biblioteca i si el llibre està disponible.
for (int i = 0; i < numLlibres; i++) {

Si es troba una coincidència, es demana a l'usuari que introdueixi el nom de la persona que agafa el llibre.
if (titols[i].equals(llibrePrestec) && !prestat[i]){
System.out.print("\nIntrodueix el nom de la persona que agafa el llibre: ");
String persona = scanner.nextLine();

Aquesta informació es guarda a l'array usuaris a la mateixa posició que el llibre. A més, es marca el llibre com prestat i es registra el préstec incrementant el comptador a numPrestecs[i].
prestat[i] = true;
usuaris[i] = persona;
numPrestecs[i]++;

Un cop completat, es mostra un missatge confirmant que el llibre ha estat prestat a la persona especificada, i la variable llibreTrobat es canvia a true. Es fa servir un break per sortir del bucle immediatament després de processar el préstec.
System.out.println("\nEl llibre ha estat prestat a " + persona);
llibreTrobat = true;
break;

Si el bucle acaba sense trobar un llibre disponible amb el títol especificat es mostra un missatge indicant que el llibre no està disponible per al préstec. Finalment, el break fora del bloc assegura que el programa surti del switch un cop s'hagi processat l'operació.
if (!llibreTrobat) {
System.out.println("\nEl llibre no està disponible per préstec.");
}
break;

Case 4
Aquest case és per retornar un llibre prestat al sistema. Comença demanant a l'usuari que introdueixi el títol del llibre que vol tornar, que es llegeix amb el scanner i es guarda a la variable llibreRetorn. Per fer un seguiment de si el llibre s'ha trobat i retornat correctament, es declara la variable llibreRetornat, inicialitzada com false.

```
System.out.print("\nIntrodueix el títol del llibre per tornar-lo: ");
scanner.nextLine();
String llibreRetorn = scanner.nextLine();
boolean llibreRetornat = false;
```

A continuació, s'utilitza un bucle for per recórrer l'array titols i buscar el llibre introduït.

```
for (int i = 0; i < numLlibres; i++) {
```

La condició del if comprova que el títol del llibre a l'array coincideix amb el títol introduït ignorant majúscules i minúscules i si el llibre està prestat.

```
if (titols[i].equalsIgnoreCase(llibreRetorn) && prestat[i]) {
```

Si aquestes condicions es compleixen, es marca el llibre com a disponible i es buida el camp de l'usuari associat. Es mostra un missatge confirmant que el llibre ha estat retornat i es canvia llibreRetornat a true. Amb el bucle amb break per evitar més iteracions innecessàries.

```
prestat[i] = false;
usuaris[i] = "";
System.out.println("\nEl llibre ha estat retornat.");
llibreRetornat = true;
break;
```

Després del bucle, si llibreRetornat continua sent false, significa que el llibre no s'ha trobat o que no estava prestat.

```
if (!llibreRetornat) {
System.out.println("\nEl llibre no està prestat.");
}
break;
```

Case 5
Aquest cas gestiona l'opció de consultar l'històric de préstecs d'un llibre. Comença demanant a l'usuari que introdueixi el títol del llibre del qual vol consultar l'històric, que es llegeix amb el scanner i s'emmagatzema a la variable llibreHistòric. Per determinar si el llibre es troba al sistema, es declara la variable booleana llibreHistoricTrobat, inicialitzada com false.

```
System.out.print("\nIntrodueix el títol del llibre per consultar l'històric de préstecs: ");
scanner.nextLine();
String llibreHistòric = scanner.nextLine();
boolean llibreHistoricTrobat = false;
```

A continuació, s'utilitza un bucle for per recórrer l'array titols i buscar si el llibre existeix a la biblioteca. Comprovem si el títol introduït coincideix amb el títol d'algun llibre registrat.

```
for (int i = 0; i < numLlibres; i++) {
if (titols[i].equalsIgnoreCase(llibreHistòric)) {
```

Si es troba una coincidència es mostra un missatge indicant quantes vegades s'ha prestat el llibre, accedint al valor guardat a l'array numPrestecs[i]. Es canvia llibreHistoricTrobat a true per indicar que s'ha trobat el llibre.

```
System.out.println("El llibre " + titols[i] + " s'ha prestat " + numPrestecs[i] + " vegades.");
llibreHistoricTrobat = true;
break;
```

Si el bucle acaba i llibreHistoricTrobat continua sent false, significa que el llibre no existeix a la biblioteca.

```
if (!llibreHistoricTrobat) {
System.out.println("\nEl llibre no es troba a la biblioteca.");
}
break;
```

Case 6

Aquest cas coincideix amb la opció de mostrar els 5 llibres més sol·licitats a la biblioteca, ordenats segons el nombre de préstecs.

Primer, es mostra un títol i es crea un array auxiliar, topPrestecs, com una còpia de l'array numPrestecs utilitzant el mètode .clone(). Això permet manipular els valors dels préstecs sense modificar l'array original.

```
System.out.println("\nLlibres més sol·licitats (Top 5):");
int[] topPrestecs = numPrestecs.clone();
```

Després, s'utilitza un bucle extern que s'executa fins a 5 vegades (una per cada llibre del top 5).

```
for (int i = 0; i < 5; i++) {
```

Dins d'aquest bucle es defineixen les variables maxPrestec i maxIndex, inicialitzades a -1. Aquestes serviran per emmagatzemar el màxim valor de préstecs i la seva posició dins de l'array durant cada iteració.

```
int maxPrestec = -1;
int maxIndex = -1;
```

Utilitzem un segon bucle intern recorre tots els llibres registrats (numLlibres) per trobar el llibre amb més préstecs:

```
for (int j = 0; j < numLlibres; j++) {
```

Si el valor de topPrestecs[j] és més gran que maxPrestec, es guarden aquest valor com a maxPrestec i la seva posició com maxIndex.

```
if (topPrestecs[j] > maxPrestec) {
maxPrestec = topPrestecs[j];
maxIndex = j;
```

Un cop finalitzat el bucle intern si maxPrestec continua sent -1, significa que ja no queden llibres per analitzar, i es trenca el bucle extern amb break.

```
if (maxPrestec == -1) break;
```

Si s'ha trobat un llibre, es mostra el títol corresponent (titols[maxIndex]) i el nombre de préstecs (maxPrestec) al top. A continuació, es marca aquest llibre com processat assignant-li un valor de -1 a topPrestecs[maxIndex], evitant que torni a ser seleccionat en les següents iteracions.

System.out.println(titols[maxIndex] + " - " + maxPrestec + " préstecs");
topPrestecs[maxIndex] = -1;

El bucle extern repeteix aquest procés fins que s'han identificat els 5 llibres més sol·licitats o fins que ja no queden llibres amb préstecs registrats.

Case 7
Aquest cas és simplement per sortir del programa.
System.out.println("\nSortint del programa...");
scanner.close();
return;

Case default
I el cas default serveix per si l'usuari s'equivoca premen una opció incorrecte.
System.out.println("\nOpció no vàlida. Torna a intentar-ho.")
break;

Activitat 3 - Joc del Buscaminas amb arrays unidimensionals

Declarem el metode main iniciem els objectes scanner i random i creem una funcio bolean per jugar de nou al joc.
Scanner scanner = new Scanner(System.in);
Random random = new Random();
boolean jugarDeNou = true;

Creem l'inici del bucle principal que controla el joc del Buscaminas. Es manté dins del bucle mentre la variable jugarDeNou mentre sigui true, cosa que ens permet repetir partides consecutives.

Configuració del tauler, el programa demana a l'usuari que introdueixi la llargada del tauler amb un valor mínim de 5, l' entrada es llegeix amb scanner i es guarda a la variable llargadaTauler.
while (jugarDeNou) {
System.out.println("\nBenvingut al joc del Buscaminas!");
System.out.print("Introdueix la llargada del tauler (mínim 5): ");
int llargadaTauler = scanner.nextInt();

Validem que la llargada del tauler sigui com a mínim 5 utilitzant un bucle while. Si el valor introduït és menor, es mostra un missatge informatiu i es demana que l'usuari introdueixi un nou valor, actualitzant la variable llargadaTauler fins que sigui vàlid.
while (llargadaTauler < 5) {
System.out.print("El tauler ha de tenir almenys 5 caselles. Torna a introduir: ");
llargadaTauler = scanner.nextInt();

Demanem el nombre de mines i guardem el valor a la variable numMines amb l'scanner.
System.out.print("Introdueix el nombre de mines (mínim 1, màxim " + (llargadaTauler - 1) + "): ");
int numMines = scanner.nextInt();

Ens assegurem que el nombre de mines sigui entre 1 i inferior a la llargada del tauler. Si no és vàlid, es mostra un missatge i es demana un nou valor fins que sigui correcte.

```
while (numMines < 1 || numMines >= llargadaTauler) {
System.out.print("Nombre invàlid de mines. Torna a introduir: ");
numMines = scanner.nextInt();
```

Inicialitzem dos arrays de caràcters amb una mida basada en la llargada del tauler. L'array mines es farà servir per emmagatzemar la posició de les mines en el tauler, mentre que l'array taulerJugador representarà el tauler visible per al jugador.

```
char[] mines = new char[llargadaTauler];
char[] taulerJugador = new char[llargadaTauler];
```

Mitjançant un bucle for, es recorre cada posició de l'array amb una mida determinada per llargadaTauler. A cada iteració, s'assigna el caràcter '-' a taulerJugador, representant les caselles no descobertes al tauler del jugador, i el caràcter '0' a mines.

```
for (int i = 0; i < llargadaTauler; i++) {
taulerJugador[i] = '-';
mines[i] = '0';
```

El codi col·loca aleatòriament mines al tauler fins que es compleixi el nombre total de mines. La variable minesSituades fa un seguiment de quantes mines s'han col·locat. Amb un bucle while generem una posició aleatòria dins del tauler utilitzant random. Si en aquesta posició no hi ha ja una mina s'hi col·loca una mina ('M') i es comptabilitza incrementant minesSituades. Aquest procés assegura que no es repeteixin posicions amb mines.

```
int minesSituades = 0;
while (minesSituades < numMines) {
int posicio = random.nextInt(llargadaTauler);
if (mines[posicio] != 'M') {
mines[posicio] = 'M';
minesSituades++;
```

Amb dues variables controlem l'estat del joc. La variable jocActiu és un booleà que indica si el joc està en marxa o s'ha acabat. La variable casellesRevelades és un comptador que registra quantes caselles han estat descobertes pel jugador.

```
boolean jocActiu = true;
int casellesRevelades = 0;
```

Iniciem el bucle principal del joc que es manté actiu mentre la variable jocActiu sigui true. Dins del bucle es mostra l'estat actual del tauler a l'usuari. Utilitzant un bucle for-each, cada casella de l'array taulerJugador es recorre i es mostra amb un espai entre elles. Això permet al jugador veure les caselles descobertes i les encara no revelades en cada torn.

```
while (jocActiu) {
for (char casella : taulerJugador) {
System.out.print(casella + " ");
```

Demanem al jugador que seleccioni una casella del tauler. Primer es mostra un missatge indicant el rang de caselles disponibles i després es llegeix la selecció de

l'usuari amb el scanner i es guarda a la variable seleccio.
System.out.print("Selecciona una casella (de 0 a " + (llargadaTauler - 1) + "): ");
int seleccio = scanner.nextInt();

Validem que la selecció sigui vàlida dins el tauler. Si la seleccio es menor que 0 o mes gran o igual que la llargada del rauler es considera invalida.
while (seleccio < 0 || seleccio >= llargadaTauler) {
System.out.print("Posició invàlida. Torna a introduir: ");
seleccio = scanner.nextInt();

Si el jugador selecciona una mina s'acaba la partida i es mostra la solució del taulell.
if (mines[seleccio] == 'M') {
System.out.println("\nBOOM! Has perdut. Hi havia una mina.");
jocActiu = false;

PEr mostrar totes les mines.
for (char casella : mines) {
System.out.print(casella + " ");

Si el jugador selecciona una posició sense mina passem a comptar mines al voltant de la casella seleccionada
int minesAlVoltant = 0;
for (int i = seleccio - 3; i <= seleccio + 3; i++) {
if (i >= 0 && i < llargadaTauler && mines[i] == 'M') {
minesAlVoltant++;

I actualitzer el tauler del jugador
taulerJugador[seleccio] = (char) ('0' + minesAlVoltant);
casellesRevelades++;

Comprovem si el jugador ha guanyat
if (casellesRevelades == llargadaTauler - numMines) {
System.out.println("\nFelicitats! Has descobert totes les caselles segures. Has guanyat!");
jocActiu = false;

I mostrem totes les mines
for (char casella : mines) {
System.out.print(casella + " ");

Preguntem si es vol tornar a jugar, si la resposta es si tornem a iniciar el joc i si no donem les gràcies per jugar.
System.out.print("Vols tornar a jugar? (s/n): ");
String resposta = scanner.nextLine().toLowerCase();
jugarDeNou = resposta.equals("s");
System.out.println("Gràcies per jugar! Fins aviat!");
scanner.close();