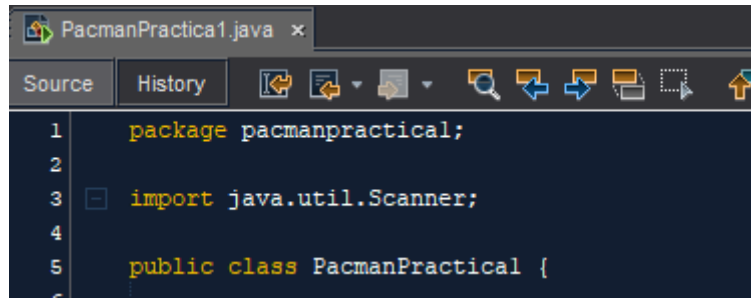


MANUAL TECNICO

13-02-2022
PACMAN IPC1

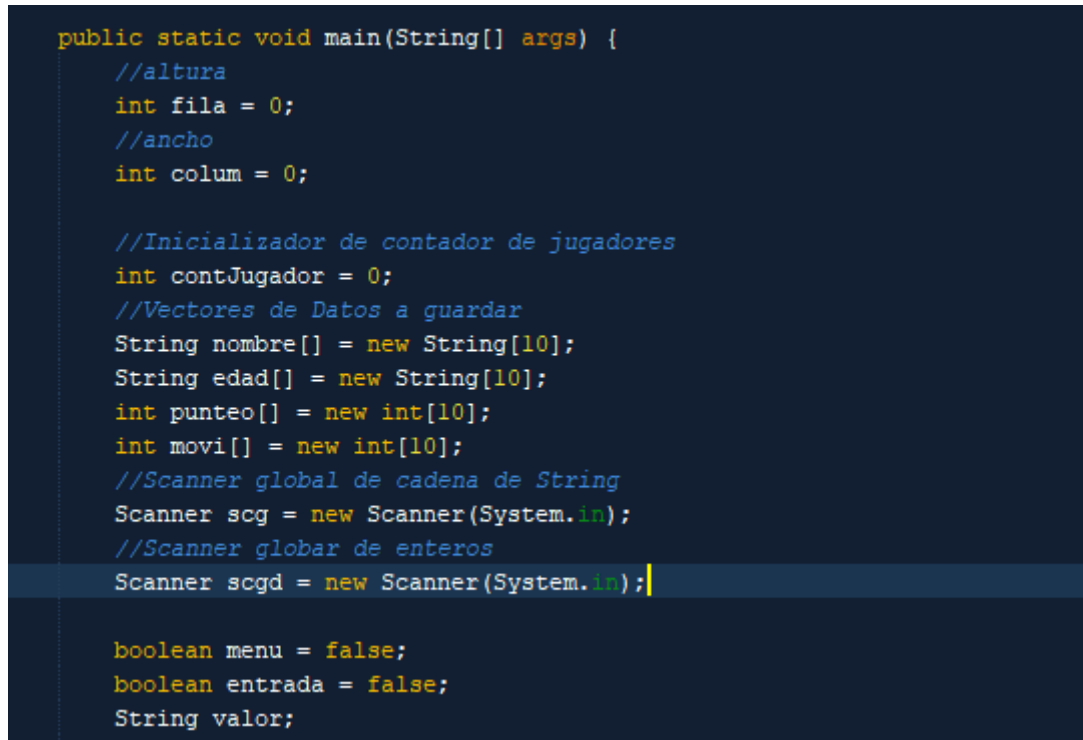
LUSVIN ALEXANDER
SICAJA RAMÍREZ
201602630

1. CLASE PRINCIPAL



```
PacmanPractica1.java x
Source History
1 package pacmanpractical;
2
3 import java.util.Scanner;
4
5 public class PacmanPractical {
6
```

Se crea la clase principal con el nombre de PacmanPractica1.java, la cual será la única que contendrá todo el código del programa PACMAN.



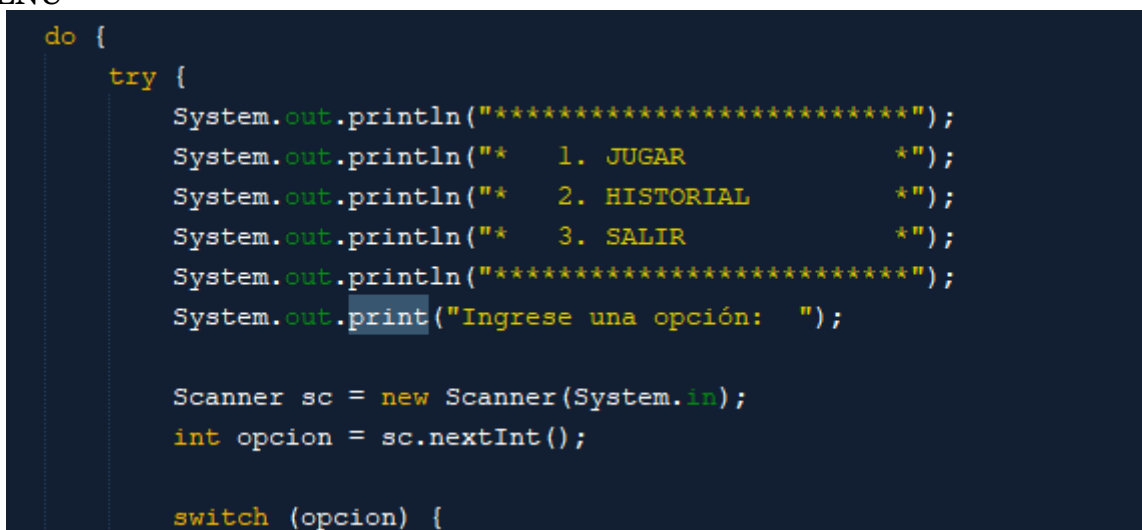
```
public static void main(String[] args) {
    //altura
    int fila = 0;
    //ancho
    int colum = 0;

    //Inicializador de contador de jugadores
    int contJugador = 0;
    //Vectores de Datos a guardar
    String nombre[] = new String[10];
    String edad[] = new String[10];
    int punteo[] = new int[10];
    int movi[] = new int[10];
    //Scanner global de cadena de String
    Scanner scg = new Scanner(System.in);
    //Scanner global de enteros
    Scanner scgd = new Scanner(System.in);

    boolean menu = false;
    boolean entrada = false;
    String valor;
}
```

Tendrá el método MAIN el cual iniciará con la aplicación mostrando los comentarios en cada variable declarada.

2. MENU



```
do {
    try {
        System.out.println("*****");
        System.out.println("* 1. JUGAR *");
        System.out.println("* 2. HISTORIAL *");
        System.out.println("* 3. SALIR *");
        System.out.println("*****");
        System.out.print("Ingrese una opción: ");

        Scanner sc = new Scanner(System.in);
        int opcion = sc.nextInt();

        switch (opcion) {

```

El menú principal estará dentro de un DO WHILE que validará siempre que la opción sea falsa para mantenerse en ejecución y al momento de ésta ser verdadera la aplicación terminará.

```
case 1:
    //Recibiendo los datos del jugador
    System.out.print("Ingrese Nombre del Jugador: ");
    nombre[contJugador] = scg.nextLine();
    System.out.print("Ingrese Edad del Jugador: ");
    edad[contJugador] = scg.nextLine();

    //Recibiendo las dimensiones del tablero de juego
    System.out.println("Ingrese las dimensiones del tablero mayor a 8x8...");
    do {
        System.out.print("alto: ");
        fila = scgd.nextInt();
        System.out.print("ancho: ");
        colum = scgd.nextInt();
        if (fila < 8 || colum < 8) {
            System.out.println("Las dimensiones son muy pequeñas ingrese nuevamente...");
        } else if (fila < colum) {
            colum = fila;
        } else {
            System.out.print("Iniciando Juego...");
        }
    }
```

En la opción 1 del case, se iniciara la solicitud de datos para la partida a ejecutar. Esta validará que las opciones sean las requeridas por el programa y de no serlo volverá a solicitar las correctas.

```
    } while (fila < 8 || colum < 8);
    System.out.println("    alto: " + fila + " ancho: " + colum);

    //Inicializando Juego
    punteo[contJugador] = 10;
    movi[contJugador] = 0;
    //matriz de tablero
    int tablero[][] = new int[fila + 1][colum + 1];

    //Posición de obtaculos
    //posición de pared fila
    int paredy = (int) (Math.random() * (fila / 2) + 2);
    //posición de pared columna
    int paredx = (int) (Math.random() * (colum / 2) + 2);
    //posición de pared fila
    int pared2y = (int) (Math.random() * (fila / 2) + 2);
```

Se iniciará con la validación de datos y configuración de variables dentro de un DO WHILE.

```

do {
    System.out.println("\t Nombre:  " + nombre[contJugador]);
    System.out.println("\t Punteo:  " + punteo[contJugador]);
    System.out.println("\t Movimientos:  " + movi[contJugador]);

    //Dibujando Matriz
    for (int i = 0; i < tablero.length; i++) {
        for (int j = 0; j < tablero[i].length; j++) {

            if (tablero[i] == tablero[posy] & tablero[j] == tablero[posx]) {
                System.out.print(" V ");
            } else if (tablero[i] == tablero[hashy] & tablero[j] == tablero[hashx]) {
                System.out.print(" # ");
            } else if (tablero[i] == tablero[dolary] & tablero[j] == tablero[dolarx]) {
                System.out.print(" $ ");
            } else if (tablero[i] == tablero[arrobay] & tablero[j] == tablero[arrobox]) {
                System.out.print(" @ ");

                //IMPRESION DE OBTACULOS
            } else if (tablero[i] == tablero[paredy] & tablero[j] == tablero[paredx]) {
                System.out.print(" * ");
            }
        }
    }
}

```

Dentro del DO WHILE se ejecutará la impresión del tablero y cada celda será verificada por condicionales que al cumplirse ejecutar la impresión del carácter correspondiente.

```

//IMPRESION DE OBTACULOS
} else if (tablero[i] == tablero[pared4y] & tablero[j] == tablero[pared4x]) {
    System.out.print(" * ");
} else if (tablero[i] == tablero[pared4y] & tablero[j] == tablero[pared4x+1]) {
    System.out.print(" * ");
} else if (tablero[i] == tablero[pared4y] & tablero[j] == tablero[pared4x+2]) {
    System.out.print(" * ");
} else if (tablero[i] == tablero[pared4y] & tablero[j] == tablero[pared4x+3]) {
    System.out.print(" * ");
}

```

Se imprime las opciones de obstáculos desde el código mostrado.

```

if (valor.equalsIgnoreCase("m") || punteo[contJugador] < 1 || punteo[contJugador] > 99) {
    System.out.println("\t ¡¡¡¡¡¡GAME OVER!!!!!!");
    entrada = true;
} else if (valor.equalsIgnoreCase("w")) {
    posy--;
    movi[contJugador]++;
} else if (valor.equalsIgnoreCase("s")) {
    posy++;
    movi[contJugador]++;
} else if (valor.equalsIgnoreCase("a")) {
    posx--;
    movi[contJugador]++;
} else if (valor.equalsIgnoreCase("d")) {
    posx++;
    movi[contJugador]++;
} else {
    System.out.println("Movimiento incorrecto...");
}

```

Para la validación de la tecla ingresando mediante el teclado se utilizan las condicionales IF y ELSE IF, las cuales validan el movimiento del PACMAN.

```

//Validación punteo hashta
if (tablero[posy] == tablero[hashy] & tablero[posx] == tablero[hashx]) {
    punteo[contJugador] = punteo[contJugador] - 10;
    hashy = (int) (Math.random() * fila + 1);
    hashx = (int) (Math.random() * colum + 1);
} else if (tablero[posy] == tablero[dolary] & tablero[posx] == tablero[dolarx]) {
    punteo[contJugador] = punteo[contJugador] + 15;
    dolary = (int) (Math.random() * fila + 1);
    dolarx = (int) (Math.random() * colum + 1);
} else if (tablero[posy] == tablero[arrobay] & tablero[posx] == tablero[arrobax]) {
    punteo[contJugador] = punteo[contJugador] + 10;
    arrobay = (int) (Math.random() * fila + 1);
    arrobax = (int) (Math.random() * colum + 1);
}

while (!entrada);

```

Para la validación de los punteos se realizan unas condicionales que ejecutan la sumatoria de los movimientos y la respectiva operación en el marcador del punteo, éste puede ser suma o resta dependiendo el valor del ITEM.

```

case 2:
    System.out.println(" ");
    System.out.println("\t HISTORIAL DE PUNTEO ");
    for (int i = 0; i < nombre.length; i++) {

        if (nombre[i] != null) {
            System.out.println("_____");
            System.out.println("Nombre: " + nombre[i]);
            System.out.println("Edad: " + edad[i]);
            System.out.println("Punteo: " + punteo[i]);
            System.out.println("Movimientos: " + movi[i]);
        } else {
            System.out.println(" ");
        }
    }
}

```

En el case 2: Se ejecuta el código mostrado para la impresión de los datos almacenados de todas las partidas ejecutadas mientras el programa no haya sido cerrado.

```

case 3:
    System.out.println("Saliendo de la aplicación...");
    menu = true;
    break;

default:
    System.out.println("Opción no encontrada....");
    break;

```

En la opción del case 3: únicamente se validara que la variable menú cambie a su estado booleano true, para que éste de salida al programa y se finalice.

Adicional se agrega una opción default, la cual muestra el mensaje descrito en el código, para notificar de una opción incorrecta.

```
    } catch (Exception e) {  
        System.out.println("INGRESE UNA OPCIÓN CORRECTA!!");  
    }
```

Se agrega una excepción de errores, para que evite que la aplicación finalice por una recepción incorrecta de parámetros por teclado.

```
    } while (!menu);
```

```
}
```

```
}
```

Por último , sí la opción de menú cambia a verdadera el DO WHILE finalizara, por lo que dará por cerrado la aplicación y eliminando los datos de las partidas guardadas.