

Explicación del Código

Paquete y Importaciones

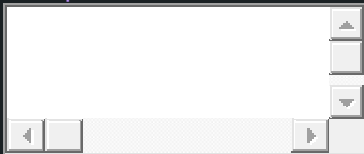
```
package juego;  
import java.util.Scanner;
```



- **package juego;**: Define el paquete al que pertenece la clase.
- **import java.util.Scanner;**: Importa la clase Scanner para leer la entrada del usuario.

Clase Principal

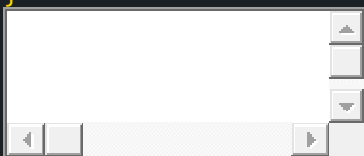
```
public class juegodepalitos {  
    private static final Scanner scanner = new Scanner(System.in);  
    private static int palitos = 21;  
    private static char simboloPalito = '|';
```



- **public class juegodepalitos**: Define la clase principal del juego.
- **private static final Scanner scanner**: Crea un objeto Scanner para leer la entrada del usuario.
- **private static int palitos = 21**: Inicializa el número de palitos a 21.
- **private static char simboloPalito = '|'**: Define el símbolo que representa a los palitos.

Método main

```
public static void main(String[] args) {  
    while (true) {  
        mostrarMenuPrincipal();  
    }  
}
```



- **public static void main(String[] args)**: Método principal que se ejecuta al iniciar el programa.
- **while (true)**: Bucle infinito que muestra el menú principal continuamente.

Método `mostrarMenuPrincipal`

```
private static void mostrarMenuPrincipal() {
    System.out.println("=====");
    System.out.println("          MENÚ PRINCIPAL          ");
    System.out.println("=====");
    System.out.println("1. Jugar");
    System.out.println("2. Controles/Ayuda");
    System.out.println("3. Salir");
    System.out.println("=====");
    System.out.print("Seleccione una opción: ");
    int opcion = scanner.nextInt();
    scanner.nextLine();
    switch (opcion) {
        case 1:
            mostrarSubmenuJugar();
            break;
        case 2:
            mostrarControlesYAyuda();
            break;
        case 3:
            System.out.println("Gracias por jugar. ¡Adiós!");
            System.exit(0);
            break;
        default:
            System.out.println("Opción no válida. Intente de nuevo.");
    }
}
```



- **`mostrarMenuPrincipal()`**: Muestra el menú principal y maneja la selección del usuario.
- **`switch (opcion)`**: Ejecuta diferentes métodos según la opción seleccionada.

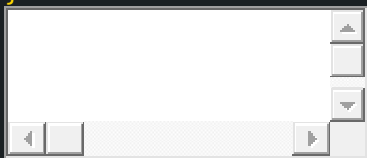
Método `mostrarSubmenuJugar`

```
private static void mostrarSubmenuJugar() {
    System.out.println("=====");
    System.out.println("          SUBMENÚ JUGAR          ");
    System.out.println("=====");
    System.out.println("1. Jugador vs. Jugador");
    System.out.println("2. Jugador vs. AI");
    System.out.println("3. Cambiar símbolo de los palitos");
    System.out.println("4. Volver al Menú Principal");
}
```

```

System.out.println("=====");
System.out.print("Seleccione una opción: ");
int opcion = scanner.nextInt();
scanner.nextLine();
switch (opcion) {
    case 1:
        jugar(false);
        break;
    case 2:
        jugar(true);
        break;
    case 3:
        cambiarSimboloPalitos();
        break;
    case 4:
        return;
    default:
        System.out.println("Opción no válida. Intente de nuevo.");
}
}

```



- **mostrarSubmenuJugar():** Muestra el submenú de juego y maneja la selección del usuario.
- **switch (opcion):** Ejecuta diferentes métodos según la opción seleccionada.

Método mostrarControlesYAyuda

```

private static void mostrarControlesYAyuda() {
    System.out.println("=====");
    System.out.println("          CONTROLES Y AYUDA          ");
    System.out.println("=====");
    System.out.println("1. El juego consiste en tachar palitos (de 1 a 4)");
    System.out.println("por turnos teniendo 21 palitos en total al inicio de la partida.");
    System.out.println("2. El jugador o máquina que tache el último palito,");
    System.out.println("pierde.");
    System.out.println("3. Puede reiniciar el juego pulsando 'R' y volver");
    System.out.println("al menú principal pulsando 'M'.");
    System.out.println("4. En el submenú de 'Jugar', puede cambiar el símbolo");
    System.out.println("que representa a los palitos.");
    System.out.println("=====");
    System.out.print("Presione Enter para volver al Menú Principal...");
    scanner.nextLine();
}

```



- **mostrarControlesYAyuda()**: Muestra las instrucciones y controles del juego.

Método jugar

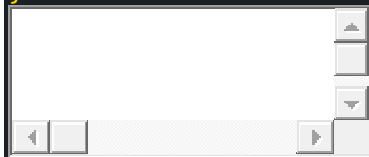
```
private static void jugar(boolean contraAI) {
    palitos = 21;
    boolean turnoJugador1 = true;
    while (palitos > 0) {
        mostrarPalitos();
        if (turnoJugador1 || !contraAI) {
            System.out.println("Turno del " + (turnoJugador1 ? "Jugador 1" : "Jugador 2") + ":");
            int palitosARemover = obtenerPalitosARemover();
            palitos -= palitosARemover;
            if (palitos <= 0) {
                System.out.println((turnoJugador1 ? "Jugador 1" : "Jugador 2") + " pierde.");
                break;
            }
        } else {
            System.out.println("Turno de la AI:\n");
            int palitosARemover = calcularMovimientoAI();
            System.out.println("AI remueve " + palitosARemover + " palitos.");
            palitos -= palitosARemover;
            if (palitos <= 0) {
                System.out.println("AI pierde.");
                break;
            }
        }
        turnoJugador1 = !turnoJugador1;
    }
    System.out.println("\n¿Desea reiniciar el juego? (R) o volver al menú principal (M): ");
    char opcion = scanner.next().charAt(0);
    if (opcion == 'R' || opcion == 'r') {
        jugar(contraAI);
    }
}
```



- **jugar(boolean contraAI)**: Maneja la lógica del juego, alternando turnos entre jugadores o contra la AI.
- **while (palitos > 0)**: Bucle que continúa hasta que no queden palitos.
- **turnoJugador1 = !turnoJugador1**: Alterna el turno entre los jugadores.

Método mostrarPalitos

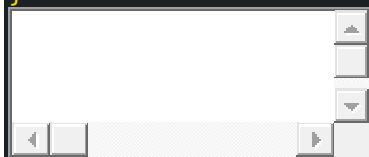
```
private static void mostrarPalitos() {  
    for (int i = 0; i < palitos; i++) {  
        System.out.print(simboloPalito);  
    }  
    System.out.println(" \n(" + palitos + " palitos restantes)\n");  
}
```



- **mostrarPalitos()**: Muestra los palitos restantes en pantalla.

Método obtenerPalitosARemover

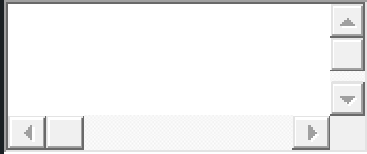
```
private static int obtenerPalitosARemover() {  
    int palitosARemover;  
    while (true) {  
        System.out.print("\nIngrese el número de palitos a remover (1-4): ");  
        palitosARemover = scanner.nextInt();  
        if (palitosARemover >= 1 && palitosARemover <= 4) {  
            break;  
        } else {  
            System.out.println("\nNúmero inválido. Intente de nuevo.");  
        }  
    }  
    return palitosARemover;  
}
```



- **obtenerPalitosARemover()**: Solicita al usuario el número de palitos a remover y valida la entrada.

Método `cambiarSimboloPalitos`

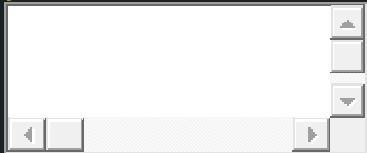
```
private static void cambiarSimboloPalitos() {  
    System.out.print("\nIngrese el nuevo símbolo para los palitos: ");  
    simboloPalito = scanner.next().charAt(0);  
    System.out.println("\nSímbolo de los palitos cambiado a: " + simboloPalito);  
}
```



- **cambiarSimboloPalitos()**: Permite al usuario cambiar el símbolo que representa a los palitos.

Método `calcularMovimientoAI`

```
private static int calcularMovimientoAI() {  
    int palitosARemover = (palitos - 1) % 5;  
    if (palitosARemover == 0) {  
        palitosARemover = 1;  
    }  
    return palitosARemover;  
}
```



- **calcularMovimientoAI()**: Calcula el movimiento óptimo para la AI.