

|  |
| --- |
|  |

METODOLOGÍA DE LA PROGRAMACIÓN

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

DRA HILDA CASTILLO ZACATELCO

PRACTICA 7

MARCO ANTONIO CIPRIAN ROMERO

//FUNCIONES SECUNDARIAS

obtenerEdad(Entrada: Entero anho Entrada: Entero mes Entrada: Entero dia Salida: Entero edad)

    Registro FechaNacimiento epoca

    Inicio

    epoca.dia <- 30, epoca.mes <- 11, epoca.anho <- 2023

    edad <- epoca.anho - anho

    Si ( (epoca.mes > mes) OR (epoca.mes = mes AND epoca.dia > dia)) Entonces

        Retonar edad

    Sino

        edad <- edad - 1

    FinSi

    Retornar edad

    Fin

validarEdad(Entrada: Registro Equipo equipo Entrada: Entero edad Salida: Booleano val )

    Inicio

    val <- Falso

        Si ( equipo.categoria = "libre" AND edad >= 18 ) Entonces

            val <- Verdadero

            Retornar val

        Sino

        Si ( equipo.categoria = "junior" AND edad >= 13 AND edad <= 17 )

            val <- Verdadero

            Retornar val

        FinSi

        Retornar val

    Fin

validarCategoria( Salida: Entero val )

    Inicio

        Escribir("Escribe la categoria del equipo: \n 1. Libre 2. Junior")

        Leer(opcion)

            Según Sea (opcion)

                Caso 1:

                    Retornar 1

                    opcion <- 0;

                Caso 2:

                    Retornar 2

                    opcion <- 0;

            Fin Según

    Fin

validarEquipo(Entrada: Cadena nombre Entrada: Entero lim Entrada: Registro Equipo equipo[], Salida: Booleano val )

    Inicio

    val <- Verdadero

    Para i <- 0 Hasta lim Incrementar 1 Hacer

        Si ( nombre = equipo[i].nombre ) Entonces

            val <- Falso

            Retornar val

        FinSi

    FinPara

    Retornar val

    Fin

validarNum(Entrada: Entero num Entrada: Registro Equipo equipo, Salida: Booleano val )

    Inicio

    val <- Verdadero

    Para i <- 0 Hasta equipo.numJugadores Incrementar 1 Hacer

        Si ( num = equipo.jugadores[i] ) Entonces

            val <- Falso

            Retornar val

        FinSi

    FinPara

    Retornar val

    Fin

indicesJg(Entrada: Entero num Entrada: Equipo equipo Salida: Entero i)

    Inicio

    i <- -1

        Mientras num <> equipo.jugadores[i].numeroJugador AND i <=9 Hacer

        i <- i + 1

        FinMientras

        Retornar i

    Fin

indicesEq(Entrada: Cadena nombre Entrada: Equipo equipo[] Salida: Entero i)

    Inicio

    i <- -1

        Mientras nombre <> equipo[i].nombre AND i <= 7 Hacer

        i <- i + 1

        FinMientras

        Retornar i

    Fin

// FUNCIONES PRINCIPALES

registrarEquipo(Entrada\Salida: Entero pos Entrada\Salida: Registro Equipo equipo[])

    Cadena nombre[30]

    Booleano val

    Inicio

    val <- Falso

        Mientras val = Falso Hacer

            Escribir("Escribe el nombre del equipo")

            Leer(nombre)

            val <- validarEquipo(nombre, pos, equipo[])

            Si ( val = Falso ) Entonces

                Escribir("Elige otro nombre para  tu equipo")

            FinSi

        FinMientras

        equipo[pos].nombre <- nombre

        pos <- pos + 1

    Fin

registrarJugador(Entrada\Salida: Registro Equipo equipo[])

    Entero pos, num, anho[4], i, dia[2], mes[2]

    Booleano val, val2

    Cadena nombre

    Inicio

    val2 <- Falso, val <- Falso

    Escribir("Escribe el nombre del equipo")

    Leer(nombre)

    i <- indicesEq(nombre, equipo[])

    pos <- equipo[i].numJugadores

    Si ( pos <= 9 AND i <= 7 ) Entonces

        Mientras val = Falso OR val2 = Falso Hacer

            Escribir("Escribe el nombre del jugador: \n nombre, apellido materno y apellido paterno")

            Leer(equipo[i].jugadores[pos].nombre)

            Escribir("Escribe el número del jugador: ")

            Leer(num)

            val <- validarNum(num, equipo)

            Si ( val = Falso ) Entonces

                Escribir("Ese número ya esta ocupado")

val2 <- Falso

            FinSi

            Escribir("Escribe la fecha de nacimiento del jugador: \n Año")

            Leer(anho)

            Escribir("Escribe la fecha de nacimiento del jugador: \n Dia")

            Leer(dia)

            Escribir("Escribe la fecha de nacimiento del jugador: \n Mes")

            Leer(mes)

            val2 <- validarEdad(equipo[i], obtenerEdad(anho,mes,dia))

            Si ( val2 = Falso ) Entonces

                Escribir("La categoría no admite esas edades")

val <- Falso

            FinSi

        FinMientras

        equipo[i].jugadores[pos].fecha.anho <- anho

        equipo[i].jugadores[pos].fecha.mes <- mes

        equipo[i].jugadores[pos].fecha.dia <- dia

        equipo[i].jugadores[pos].numeroJugador <- num

        equipo[i].numJugadores <- pos + 1

    Sino

        Escribir("Se ha alcanzado el límite de jugadores o no existe ese equipo")

    FinSi

    Si (equipo[i].numJugadores >= 6) Entonces

       equipo[i].estado <- “completo”

    Sino

       equipo[i].estado <- “incompleto”

    FinSi

    Fin

eliminarJugador( Entrada\Salida: Registro Equipo equipo[] )

    Entero num, i, k

    Cadena nombre

    Registro Jugador vacio

    Inicio

    Escribir("Ingresa el nombre del equipo")

    Leer(nombre)

    k <- indicesEq(nombre, equipo[])

    Escribir("Ingresa el número del jugador")

    Leer(num)

    i <- indicesEq(num, equipo[k])

    Si (i <=  9 AND k <= 7) Entonces

    equipo[k].jugadores[9] <- vacio

        Para j <- i Hasta 8 Incrementar 1 hacer

            equipo[k].jugadores[j] <- equipo[k].jugadores[j+1]

        FinPara

    equipo[k].numJugadores <- equipo[k].numJugadores - 1

    Sino

    Escribir("No se encontró ese jugador u equipo")

    FinSi

    Fin

eliminarEquipo(Entrada\Salida: Entero pos Entrada\Salida: Registro Equipo equipo[] )

    Entero k

    Cadena nombre

    Registro Equipo vacio

    Inicio

    Escribir("Ingresa el nombre del equipo")

    Leer(nombre)

    k <- indicesEq(nombre, equipo[])

    Si (k <= 7 AND equipo[k].numJugadores = 0) Entonces

    equipo[7] <- vacio

        Para j <- k Hasta 6 Incrementar 1 hacer

            equipo[j] <- equipo[j+1]

        FinPara

    pos <- pos -1

    Sino

    Escribir("No se encontró ese equipo, no puedes borrar equipos con jugadores")

    FinSi

    Fin

mostrarEquipo(Entrada: Registro Equipo equipo[] )

    Entero num, k

    Cadena nombre

    Registro Equipo vacio

    Inicio

    Escribir("Ingresa el nombre del equipo")

    Leer(nombre)

    k <- indicesEq(nombre, equipo[])

    Si (k <= 7) Entonces

        Escribir("Nombre: ",  equipo[k].nombre, "Estado: ", equipo[k].estado, "\n Jugadores: \n

        Nombre | Número |")

        Para i <- 0 Hasta equipo[k].numJugadores Incrementar 1 Hacer

            Escribir(equipo[k].jugadores[i].nombre, " ", equipo[k].jugadores[i].numeroJugador)

        FinPara

    Sino

    Escribir("No se encontró ese equipo")

    FinSi

    Fin

mostrarJugador( Entrada: Registro Equipo equipo[] )

    Entero num, i, k

    Cadena nombre

    Registro Jugador jugador

    Inicio

    Escribir("Ingresa el nombre del equipo")

    Leer(nombre)

    k <- indicesEq(nombre, equipo[])

    Escribir("Ingresa el número del jugador")

    Leer(num)

    i <- indicesEq(num, equipo[k])

    jg <- equipo[k].jugadores[i]

    fn <- equipo[k].jugadores[i].fecha

    Si (i <=  9 AND k <= 7) Entonces

        Escribir("Nombre | Equipo | Nacimiento | Número")

        Escribir(jg.nombre, " ", nombre, " ", fn.dia, "/", fn.mes, "/", fn.anho, " ", jg.numeroJugador)

    Sino

    Escribir("No se encontró ese jugador u equipo")

    FinSi

    Fin

 //INICIO DEL ALGORITMO

Algoritmo

Registro FechaNacimiento =

    Entero dia, mes, anho

FinRegistro

Registro Jugador =

    Cadena nombre[50]

    Registro FechaNacimiento fecha

    Entero numeroJugador

FinRegistro

Registro Equipo =

    Cadena nombre[30], estado[10], categoria[10]

    Registro Jugador jugadores[10]

    Entero numJugadores

FinRegistro

Entero EquipoL, EquipoJ, subOpc

Inicio

    equipoL <- 0, equipoJ <- 0

    Registro Equipo libre[8], Registro Equipo junior[8]

    Para i <-0 Hasta 7 Incrementar 1 Hacer

        libre[i].categoria <- "libre"

        junior[i].categoria  <- "junior"

    FinPara

    Repetir

    Escribir("\n=== Menú ===\n")

    Escribir("1. Registrar Jugador\n")

    Escribir("2. Registrar Equipo\n")

    Escribir("3. Eliminar Jugador\n")

    Escribir("4. Eliminar Equipo\n")

    Escribir("5. Mostrar Jugador\n")

    Escribir("6. Mostrar Equipo\n")

    Escribir("0. Salir\n")

    Escribir("Ingrese una opción: ")

    Leer(opcion)

    Según Sea (opcion)

        caso 1:

            subOpc <- validarCategoria()

            Si ( subOpc = 1 ) Entonces

                registrarJugador(libre[])

            Sino

            Si ( subOpc = 2 ) Entonces

                registrarJugador(junior[])

            FinSi

        caso 2:

            subOpc <- validarCategoria()

            Si ( subOpc = 1 AND equipoL <= 7 ) Entonces

                registrarEquipo(equipoL, libre[])

            Sino

            Si ( subOpc = 2 AND equipoJ <= 7 ) Entonces

                registrarEquipo(equipoJ, junior[])

            Sino

                Escribir("Se ha alcanzado el limite de equipos")

            FinSi

        caso 3:

            subOpc <- validarCategoria()

            Si ( subOpc = 1 ) Entonces

                eliminarJugador(libre[])

            Sino

            Si ( subOpc = 2 ) Entonces

                eliminarJugador(junior[])

            FinSi

        caso 4:

            subOpc <- validarCategoria()

            Si ( subOpc = 1 ) Entonces

                eliminarEquipo(equipoL, libre[])

            Sino

            Si ( subOpc = 2 ) Entonces

                eliminarEquipo(equipoJ, junior[])

            FinSi

        caso 5:

            subOpc <- validarCategoria()

            Si ( subOpc = 1 ) Entonces

                mostrarEquipo(libre[])

            Sino

            Si ( subOpc = 2 ) Entonces

                mostrarEquipo(junior[])

            FinSi

        caso 6:

            subOpc <- validarCategoria()

            Si ( subOpc = 1 ) Entonces

                mostrarJugador(libre[])

            Sino

            Si ( subOpc = 2 ) Entonces

                mostrarJugador(junior[])

            FinSi

        caso 0:

            printf("Saliendo del programa. ¡Hasta luego!\n");

        caso contrario:

            printf("Opción no válida. Intente nuevamente.\n");

    FinSegún

    Hasta (opcion = 0)

Fin