

PROCÉS DE PROGRAMACIÓ

1. Estructura principal del programa

```
package PE02.src;
import java.util.Scanner;
import java.time.Year;
import java.text.DecimalFormat;
import java.util.Locale;

public class prova {
    Run | Debug
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in).useLocale(Locale.US);
        DecimalFormat df2 = new DecimalFormat(pattern: "#.##");
        DecimalFormat df0 = new DecimalFormat(pattern: "#");
        char choice;
        boolean program = true;
        String fullName;
        int age;
        double weight, height;

        while (program){
            System.out.println(x:"--- CONTROL DE SALUT ---");
            System.out.println(x:"a) Introduir dades");
            System.out.println(x:"b) Modificar dades");
            System.out.println(x:"c) Visualitzar dades");
            System.out.println(x:"d) Sortir");
            System.out.print(s:"Tria una opcio: ");
            choice = scanner.next().toLowerCase().charAt(index:0);

            switch (choice) {
                case 'a':
                    System.out.println(x:"Has triat introduir dades.");
                    // Code to enter data
                    break;

                case 'b':
                    System.out.println(x:"Has triat modificar dades.");
                    // Code to modify data
                    break;

                case 'c':
                    System.out.println(x:"Has triat visualitzar dades.");
                    // Code to visualize data
                    break;

                case 'd':
                    System.out.println(x:"Sortint del programa.");
                    program = false;
                    break;

                default:
                    System.out.println(x:"Opcio no valida. Si us plau, tria una opcio entre a i d.");
            }
        }
    }
}
```

2. Completació del case a

```
// NOM COMPLET
boolean validName = false;
while (!validName) {
    try {
        System.out.print(s:"Nom complet: ");
        fullName = scanner.nextLine().trim(); // Trim per netejar espais sobrants
        if (fullName.isEmpty()) {
            throw new Exception(message:"Error: El nom no pot quedar buit.");
        }
        validName = true; // Si arriba aquí nom = valid
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
```

```
// EDAT
boolean validAge = false;
while (!validAge) {
    try {
        System.out.print(s:"Edat: ");
        String ageInput = scanner.nextLine().trim();
        age = Integer.parseInt(ageInput); // Si dona error vol dir que hi ha error d'entrada
        if (age <= 0 || age > 120) {
            throw new Exception(message:"Error: L'edat ha de ser un enter positiu < 0 = a 120.");
        }
        validAge = true;
    } catch (NumberFormatException e) {
        System.out.println(x:"Error: Format numèric invàlid.");
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
```

```
// PES
boolean validWeight = false;
while (!validWeight) {
    try {
        System.out.print(s:"Pes (kg): ");
        String weightInput = scanner.nextLine().trim().replace(oldChar:',', newChar:'.'); // Accepta . i , com a decimals
        weight = Double.parseDouble(weightInput);
        if (weight <= 0 || weight > 400) {
            throw new Exception(message:"Error: El pes ha de ser un decimal positiu < 400.");
        }
        validWeight = true;
    } catch (NumberFormatException e) {
        System.out.println(x:"Error: Format numèric invàlid.");
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
```

```
// ALTURA
boolean validHeight = false;
while (!validHeight) {
    try {
        System.out.print(s:"Alçada (m): ");
        String heightInput = scanner.nextLine().trim().replace(oldChar:',', newChar:'.');
        height = Double.parseDouble(heightInput);
        if (height < 0.5 || height > 2.5) {
            throw new Exception(message:"Error: L'alçada ha de ser un decimal positiu entre 0.5 i 2.5 metres.");
        }
        validHeight = true;
    } catch (NumberFormatException e) {
        System.out.println(x:"Error: Format numèric invàlid.");
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}
```

3. Comprovar que s'han introduït dades abans de modificar o visualitzar

dins del case a

```
enteredData = true;
```

dins del case b i c

```
if (!enteredData) {
    System.out.println(x:"Error: Primer has d'introduir les dades (opció a).");
    break;
}
```

4. Estructura del case b

```

case 'b':
    if (!enteredData) {
        System.out.println(x:"Error: Primer has d'introduir les dades (opció a).");
        break;
    }

    System.out.println(x:"Has triat modificar dades.");
    System.out.println(x:"1) Nom complet");
    System.out.println(x:"2) Edat");
    System.out.println(x:"3) Pes");
    System.out.println(x:"4) Alçada");
    System.out.print(s:"Quina dada vols modificar? (1-4): ");
    String modifyInput = scanner.nextLine().trim();

    try {
        int modifyChoice = Integer.parseInt(modifyInput); //S'agafa com a string per controlar errors i es converteix a int

        switch(modifyChoice) {
            // NOM
            case 1:
                break;
            // EDAT
            case 2:
                break;
            // PES
            case 3:
                break;
            // ALTURA
            case 4:
                break;
            default:
                System.out.println(x:"Error: Opció no vàlida.");
        }
    } catch (NumberFormatException e) {
        System.out.println(x:"Error: Format numèric invàlid.");
    }
    break;

```

5. Completat case 1, 2, 3 i 4

```

case 1:
    boolean validNameMod = false;
    while (!validNameMod) {
        try {
            System.out.print(s:"Nou nom complet: ");
            String newName = scanner.nextLine().trim(); // Trim per netejar espais sobrants
            if (newName.isEmpty()) {
                throw new Exception(message:"Error: El nom no pot quedar buit.");
            }
            fullName = newName; // Assignar el nou nom si és vàlid
            validNameMod = true; // Si arriba aquí nom = valid
            System.out.println(x:"Nom modificat correctament");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
    break;

```

```

case 2:
    boolean validAgeMod = false;
    while (!validAgeMod) {
        try {
            System.out.print(s:"Nova edat: ");
            String newAgeInput = scanner.nextLine().trim();
            int newAge = Integer.parseInt(newAgeInput); //Si dona error vol dir que hi ha error d'entrada
            if (newAge <= 0 || newAge > 120) {
                throw new Exception(message:"Error: L'edat ha de ser un enter positiu < 0 = a 120.");
            }
            age = newAge; // Assignar la nova edat si és vàlida
            validAgeMod = true;
            System.out.println(x:"Edat actualitzada correctament.");
        } catch (NumberFormatException e) {
            System.out.println(x:"Error: Format numèric invàlid.");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
    break;

```

```

case 3:
    boolean validWeightMod = false;
    while (!validWeightMod) {
        try {
            System.out.print(s:"Nou pes (kg): ");
            String newWeightInput = scanner.nextLine().trim().replace(oldChar:',', newChar:'.'); //Accepta . i , com a decimals
            double newWeight = Double.parseDouble(newWeightInput);
            if (newWeight <= 0 || newWeight > 400) {
                throw new Exception(message:"Error: El pes ha de ser un decimal positiu raonable.");
            }
            weight = newWeight; // Assignar el nou pes si és vàlid
            validWeightMod = true;
            System.out.println(x:"Pes actualitzat correctament.");
        } catch (NumberFormatException e) {
            System.out.println(x:"Error: Format numèric invàlid.");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
    break;

```

```

case 4:
    boolean validHeightMod = false;
    while (!validHeightMod) {
        try {
            System.out.print(s:"Nova alçada (m): ");
            String newHeightInput = scanner.nextLine().trim().replace(oldChar:',', newChar:'.');
            double newHeight = Double.parseDouble(newHeightInput);
            if (newHeight < 0.5 || newHeight > 2.5) {
                throw new Exception(message:"Error: L'alçada ha de ser un decimal positiu entre 0.5 i 2.5 metres.");
            }
            height = newHeight; // Assignar la nova alçada si és vàlida
            validHeightMod = true;
            System.out.println(x:"Alçada actualitzada correctament.");
        } catch (NumberFormatException e) {
            System.out.println(x:"Error: Format numèric invàlid.");
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
    break;

```

6. Operacions per visualitzar les dades en el case c

```
// Normalitzar el NOM
String normalizedName = "";
String[] words = fullName.trim().split(regex:" "); // Dividir el nom en paraules (per espais)
for (int i = 0; i < words.length; i++) { // Processar cada paraula
    if (!words[i].isEmpty()) { // Primera lletra en majúscula, la resta en minúscula
        String firstLetter = words[i].substring(beginIndex:0, endIndex:1).toUpperCase();
        String restOfWord = words[i].substring(beginIndex:1).toLowerCase();
        String normalizedWord = firstLetter + restOfWord;
        if (normalizedName.isEmpty()) { // Afegir al resultat
            normalizedName = normalizedWord;
        } else {
            normalizedName = normalizedName + " " + normalizedWord;
        }
    }
}
}
```

```
// Càlcul IMC
double imc = weight / (height * height);
String IMC_category = "";
if (imc < 18.5) {
    IMC_category = "pes baix";
} else if (imc < 25) {
    IMC_category = "pes normal";
} else if (imc < 30) {
    IMC_category = "sobrepès";
} else {
    IMC_category = "obesitat";
}
```

```
// Freqüència cardíaca
int fc_max = 220 - age;
int fc50 = (int) Math.round(fc_max * 0.5);
int fc85 = (int) Math.round(fc_max * 0.85);
```

```
// Aigua recomanada
double waterLiters = (weight * 35) / 1000.0;
```

```
// Any de naixement
int actualYear = Year.now().getValue();
int birthYear = actualYear - age;
```

7. Imprimir les dades, inicialitzar les variables al principi i eliminar variables no utilitzades

```
// Mostrar resultats
System.out.println("Hola, " + normalizedName + "!");
System.out.println("Edat: " + age + " anys | Pes: " + df2.format(weight) + " kg | Alçada: " + df2.format(height) + " m");
System.out.println("IMC: " + df2.format(imc) + " (" + IMC_category + ")");
System.out.println("FC màxima estimada: " + fc_max + " bpm");
System.out.println("Zona FC objectiu: " + fc50 + "-" + fc85 + " bpm");
System.out.println("Aigua recomanada: " + df2.format(waterLiters) + " L/dia");
System.out.println("Any de naixement aproximat: " + birthYear);
```

```
package PE03.src;
import java.util.Scanner;
import java.time.Year;
import java.text.DecimalFormat;
import java.util.Locale;

public class test {
    Run | Debug
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in).useLocale(Locale.US);
        DecimalFormat df2 = new DecimalFormat(pattern:"#.##");
        char choice;
        boolean program = true;
        String fullName = "";
        int age = 0;
        double weight = 0;
        double height = 0;
        boolean enteredData = false;
```

PROVES

```
--- CONTROL DE SALUT ---
a) Introduir dades
b) Modificar dades
c) Visualitzar dades
d) Sortir
Tria una opció: b
Error: Primer has d'introduir les dades (opció a).
```

```
--- CONTROL DE SALUT ---
a) Introduir dades
b) Modificar dades
c) Visualitzar dades
d) Sortir
Tria una opció: c
Error: Primer has d'introduir les dades (opció a).
```

--- CONTROL DE SALUT ---

- a) Introduir dades
- b) Modificar dades
- c) Visualitzar dades
- d) Sortir

Tria una opció: a

Has triat introduir dades.

Nom complet:

Error: El nom no pot quedar buit.

Nom complet: sadads sfds

Edat: 300

Error: L'edat ha de ser un enter positiu $< 0 = a$ 120.

Edat: 10

Pes (kg): 1000

Error: El pes ha de ser un decimal positiu < 400 .

Pes (kg): 100

Alçada (m): 10

Error: L'alçada ha de ser un decimal positiu entre 0.5 i 2.5 metres.

Alçada (m): 1

Tria una opció: b

Has triat modificar dades.

1) Nom complet

2) Edat

3) Pes

4) Alçada

Quina dada vols modificar? (1-4): 3

Nou pes (kg): 100

Pes actualitzat correctament.

--- CONTROL DE SALUT ---

a) Introduir dades

b) Modificar dades

c) Visualitzar dades

d) Sortir

Tria una opció: c

Has triat visualitzar dades.

Hola, Sadads Sfds!

Edat: 10 anys | Pes: 100 kg | Alçada: 1 m

IMC: 100 (obesitat)

FC màxima estimada: 210 bpm

Zona FC objectiu: 105-179 bpm

Aigua recomanada: 3.5 L/dia

Any de naixement aproximat: 2015

--- CONTROL DE SALUT ---

a) Introduir dades

b) Modificar dades

c) Visualitzar dades

d) Sortir

Tria una opció: d

Sortint del programa.

PS C:\Users\g6thi\OneDrive\Desktop\DAM1\Programacio\practiques> |