Laços Condicionais

A estrutura condicional ajuda os desenvolvedores a economizar linha de código, fazer reaproveitamento dos scripts e auxiliam na tomada de decisão.

Essas estruturas são utilizadas quando há uma sequencia de ações que podem ou não ser executadas, dependendo do resultado de uma operação ou interação.

Por exemplo: ao utilizar o Scanner, o programa coleta a idade de um usuário, recebendo o valor age = 17. O mesmo programa, ao receber uma idade menor que 18, informa que o usuário não é maior de idade. Se a idade for maior ou igual a 18 anos, uma mensagem diferente será enviada, informando que o usuário é maior de idade.

Essas estruturas incluem o uso de if, else if, else e switch-case, que permitem a execução de diferentes blocos de acordo com as condições especificadas.

Na aula de hoje, vamos estudar sobre o Laço Condicional e aprender a criar esses laços com a ajuda de exemplos.

1.1 IF, ELSE IF, ELSE

O Laço Condicional if é utilizado para testar uma condição, em que, sendo ela verdadeira, será executado um processo. O else if e else permitem adicionar condições adicionais para executar um bloco de código caso as condições anteriores forem falsas.

Veja como é a sintaxe desse laço condicional abaixo:

if(condição1){

// Executa o código caso a condição1 seja verdadeira

} else if (Condição2) {

// Executa o código caso a condição2 seja verdadeira

} else {

// Executa o código caso a condição1 e condição2 sejam falsas

}

Vamos criar um exemplo juntos, e para começar vamos criar um fluxograma para entender a lógica envolvida no programa:

Sim

Executa Condição A

Condição A

Não

Sim

Executa Condição B

Condição B

Não

Sim

Condição C

Executa Condição C

Não

Agora veja abaixo o código que corresponde ao fluxograma que acabamos de criar:

package Condicional;

public class IfElse {

public static void main(String[] args) {

int x = 10;

if (x > 0) {

System.out.println("x é positivo");

} else if (x < 0) {

System.out.println("x é negativo");

} else {

System.out.println("x é zero");

}

}

}

No exemplo acima temos três condições, onde o programa executa o bloco de código da condição que for verdadeira entre o if, else if ou else.

1. Se x for maior que 0, a mensagem executada será: "x é positivo";
2. Se x for menor que 0, a mensagem executada será: "x é negativo";
3. Se x for menor que 0, a mensagem executada será: "x é zero".

1.2 SWITCH-CASE

A estrutura condicional com switch-case vem como alternativa para caso seja necessário utilizar diversos ifs no código. Quando utilizamos if/else, a estrutura do código pode ficar mais extensa, e em caso de manutenção, poderia nos deixar confusos.

O switch-case testará o valor contido em uma variável, fazendo uma comparação com cada uma das opções. Enquanto o valor da variável ser similar ao número da condição, o programa executar um bloco de instruções.

Veja um exemplo da sintaxe abaixo:

switch(expressão){

case Valor1:

// Exectará uma ação caso a condição seja verdadeira

break;

case Valor2:

// Executará uma ação caso a condição seja verdadeira

break;

// Outros cases...

default:

// Executará uma ação caso as condições anteriores sejam falsas

}

Vamos fazer novamente um fluxograma e um código para entendermos na prática:

O em nosso fluxograma, iremos escolher o mês de acordo com o seu número de sequência. Por exemplo: janeiro é o mês 1, fevereiro é o mês 2, e assim por diante. A proposta é o Usuário escolherá um valor e sua opção ser atendida.

Insira a opção

Caso opção == Mês

Caso opção seja diferente

Executa um bloco de código referente ao mês

Executa uma mensagem: “Opção Inválida!”

Fim

Início

Leia a opção

No fluxograma acima, o usuário deve escolher um número. Se o número for igual ao número de alguma condição, o programa executará um bloco de código respectivo a escolha do usuário.

Agora, vamos escrever o nosso código?

ppackage Condicional;

import java.util.Scanner;

public class SwitchCase {

public static void main(String[] args) {

int mes;

String nomeMes;

Scanner leia = new Scanner(System.in);

System.out.println(

"Escolha o seu mês de nascimento: "

+"\n################################################################"

+ "\n # 1 - Janeiro; # 2 - fevereiro; # 3 - Março #"

+ "\n # 4 - Abril; # 5 - Maio; # 6 - Junho #" + "\n # 7 - Julho; # 8 - Agosto; # 9 - Setembro #"

+ "\n # 10 - Outubro; # 11 - Novembro; # 12 - Dezembro #"

+ "\n################################################################"

);

mes = leia.nextInt();

leia.close();

switch (mes) {

case 1: {

nomeMes = "Janeiro";

break;

}

case 2: {

nomeMes = "Fevereiro";

break;

}

case 3: {

nomeMes = "Março";

break;

}

case 4: {

nomeMes = "Abril";

break;

}

case 5: {

nomeMes = "Maio";

break;

}

case 6: {

nomeMes = "Junho";

break;

}

case 7:{

nomeMes = "Julho";

break;

}

case 8: {

nomeMes = "Agosto";

break;

}

case 9: {

nomeMes = "Setembro";

break;

}

case 10: {

nomeMes = "Outubro";

break;

}

case 11: {

nomeMes = "Novembro";

break;

}

case 12: {

nomeMes = "Dezembro";

break;

}

default:

throw new IllegalArgumentException(

"O mês selecionado é inválido: " + mes);

}

System.out.println(

"\n Você me informou que nasceu no mês de " + nomeMes);

}

}

No código acima, foi criado um objeto Scanner para ler a opção que o usuário deseja, esta opção foi atribuída a variável mês que será comparada no switch-case.

Caso a condição seja atendida em um dos cases, a variável nomeMes será retornada com o correspondente valor selecionado. Caso a condição não seja atendida, uma mensagem de erro será emitida.

1.3 Quando utilizar a estrutura if/else e switch/case

Podemos utilizar ‘if’, ‘else if’, ‘else’ quando há uma condição booleana que precisa ser avaliada. É adequado para casos onde as condições são complexas e não podem ser facilmente reduzidas a uma única expressão.

Utilizamos ‘switch-case’ para cenários onde uma única variável precisa ser comparada com diversos valores constantes. A tendencia do ‘switch-case’ é ser mais legível e eficiente do que uma longa cadeia de ‘if-else’.

Conclusão

Utilizamos o Laço Condicional para determinar quais ações serão realizadas para cada situação ou condição diferente. Fique à vontade para treinar, fazer exercícios, e quem sabe juntar com os conceitos que estudamos nas aulas passadas.

Referência

Estruturas condicionais e estruturas de repetição em Java. Acessado dia 03 de Junho de 2024.

https://www.treinaweb.com.br/blog/estruturas-condicionais-e-estruturas-de-repeticao-em-java#google\_vignette

Laços Condicionais. Acessado dia 03 de Junho de 2024.

https://github.com/Leon4rdoalves/CookBook-Java/blob/main/06.md