# Aula 04-Desvios condicionais encadeados

## Introdução

Olá, meus estudantes, sejam bem-vindos à nossa aula de Algoritmo e Programação em Java do Instituto da Oportunidade Social. Nessa aula, você vai conhecer como implementar desvios condicionais encadeados.

## Desvio condicional encadeado

Muitas vezes, em programação, precisamos testar diversas condições e cada condição depende do resultado da condição, que foi testada anteriormente. Esse tipo de desvio condicional também pode ser chamado de ifs aninhados, que basicamente é colocar um if dentro de if.

Diagram

Description automatically generated

A sintaxe de um desvio condicional encadeado é:

Text, letter

Description automatically generated

Fique atento as aberturas e fechamento de chaves para cada bloco do desvio condicional encadeado.

### Vamos programar

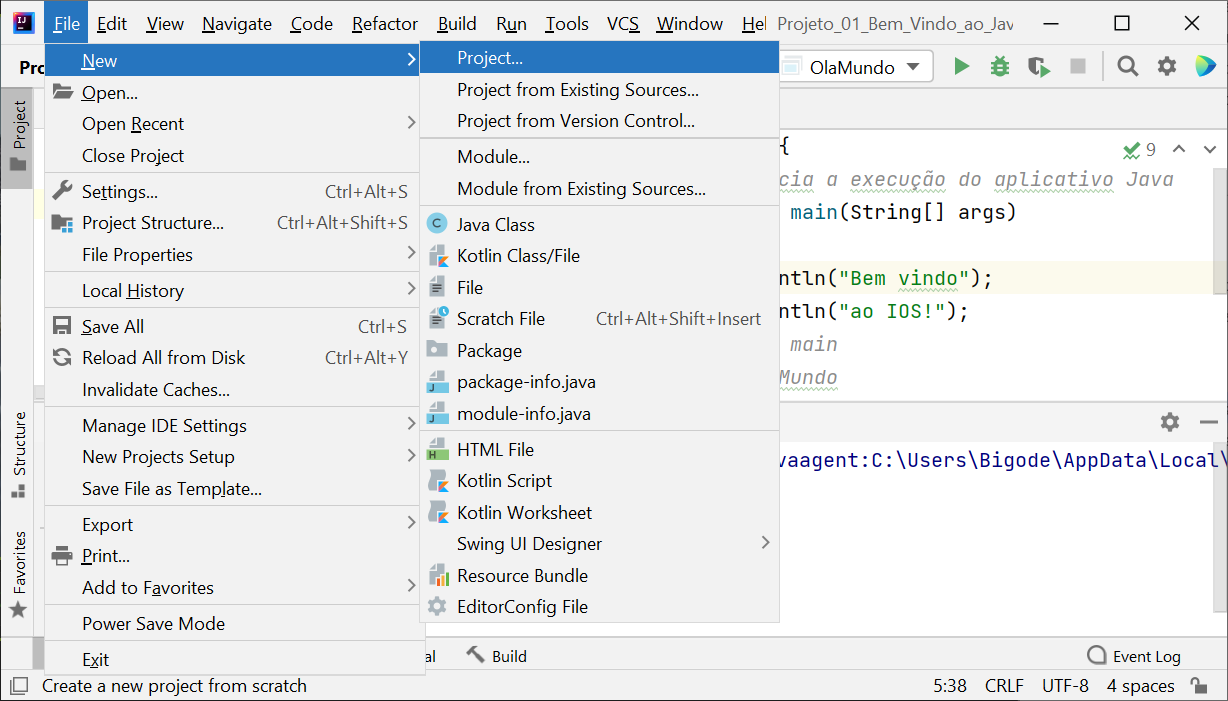
Vamos fazer um programa do Jogo de adivinhação. O objetivo do jogo é adivinhar o número 15, predeterminado no programa. O usuário irá digitar

Diagram

Description automatically generated

Portanto, siga os passos para escrever o nosso programa para o jogo de adivinhação:

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.



1. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_08\_Adivinha
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: JogoAdivinha e insira o código mostrado na Figura 1.

|  |
| --- |
| 1. **import** java.util.Scanner; // Import necessário para utilizar a classe 3. **public** **class** JogoAdivinha { 4. **public** **static** **void** main (String [] args) { 5. // Intância do objeto (buffer) utilizando a classe Scanner 6. Scanner entrada = **new** Scanner(System.in); 7. **int** palpite; 8. System.out.println("Digite seu palpite:"); 9. palpite = entrada.nextInt(); 10. **if**(palpite == 15) { 11. System.out.println("Você acertou!!!"); 12. } 13. **else** { 14. **if**(palpite < 15) { 15. System.out.println("Seu palpite está abaixo !!!"); 16. } **else** { 17. System.out.println("Seu palpite está acima."); 18. } 19. } 20. entrada.close(); 21. } // fim do método main 22. } // fim da classe JogoAdivinha |

Figura - Código do programa para o jogo de adivinhação.

1. Ao executar o programa, ele irá solicitar que o usuário digite um palpite e então o programa testará as condições dos if encadeados e imprimirá uma das mensagens de acordo com a condição verdadeira.

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

A instrução na linha 10 da Figura 1 verifica se o palpite digitado é igual a 15. Caso verdadeiro a condição o programa exibirá a mensagem que você acertou. Caso a condição seja falsa, o programa irá verificar a condição do if na linha 14, caso seja verdadeira, ele imprimirá que o palpite está abaixo do valor 15. Caso a condição seja falsa, o programa executa o bloco de comandos do else na linha 17 e imprimirá a mensagem que o palpite está acima do valor 15.

Vamos deixar esse programa um pouco mais difícil utilizando a classe **Random** para gerar um número aleatório de 1 a 10 para que possamos tentar adivinhar.

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.

Graphical user interface, text, application, email

Description automatically generated

1. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_09\_Adivinha\_Random
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: AdivRandom e insira o código mostrado na Figura 2.

|  |
| --- |
| 1. **import** java.util.Scanner; // Import necessário para utilizar a classe Scanner 2. **import** java.util.Random;  // Import necessário para utilizar a classe Random 4. **public** **class** AdivRandom { 5. **public** **static** **void** main (String [] args) { 6. // Intância (buffer) utilizando a classe Scanner 7. Scanner entrada = **new** Scanner(System.in); 8. // Intância utilizando a classe Random 9. Random rand = **new** Random(); 10. **int** palpite, num = 0; 11. num = rand.nextInt(10) + 1; 12. System.out.println("Digite seu palpite:"); 13. palpite = entrada.nextInt(); 14. **if**(palpite == num) { 15. System.out.println("Você acertou!!! O número era " + num); 16. } 17. **else** { 18. **if**(palpite < num) { 19. System.out.println("Seu palpite está abaixo!!! O número era " + num); 20. } **else** { 21. System.out.println("Seu palpite está acima. O número era " + num); 22. } 23. } 24. entrada.close(); 25. } // fim do método main 26. } // fim da classe AdivRandom |

Figura 2 - Código do programa para o jogo de adivinhação com a classe Random.

1. Ao executar o programa, ele irá solicitar que o usuário digite um palpite e então o programa testará as condições dos if encadeados e imprimirá uma das mensagens de acordo com a condição verdadeira.

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

A instrução da linha 9 criar uma instância da classe Random, que nos permite “sortear” um valor aleatório. Na linha 11, o método nextInt(10) sorteia um valor inteiro de 0 a 9, por isso precisamos somar 1 para que a faixa dos números possíveis a serem sorteados fiquem entre 1 e 10.

## Switch-case

O comando **switch-case** é utiliza para trabalhar com situações mutualmente exclusivas. Situações mutualmente exclusivas são aquelas que se uma situação for executada, as demais não poderão ser executadas. O **switch-case** sempre verifica o valor de uma **expressão** para decidir o que fazer, ou seja, qual(is) instrução(ões) será(ão) verificada(s). A **expressão** será comparada com o valor de cada comando **case**. Sempre que a **expressão** casar com o valor de um **case** a lista de comando desse **case** é executada e o **switch** finaliza ao encontrar o comando **break**. A **expressão** comparada pode ser do tipo **int**, **char** ou **String**. A sintaxe do **switch-case** é:

Diagram

Description automatically generated

O caso **default** é executado quando nenhum **case** for satisfeito. O comando **switch** não pode ter dois ou mais comandos **case** com valores iguais.

Se compararmos o **switch-case** com os ifs encadeados veremos que eles têm muitas semelhanças.

Graphical user interface, text, application

Description automatically generated

### Vamos praticar

Vamos criar um programa para exibir um menu de opções para o usuário escolher. O programa inicialmente deverá mostrar o seguinte texto na tela:

Text

Description automatically generated

O Usuário deverá digitar uma das três opções:

|  |  |
| --- | --- |
| Caso digite 1 deverá aparecer:   * “Você escolheu o menu 1” * “Que é a opção Cadastrar Produtos” | Caso digite 2 deverá aparecer:   * “Você escolheu o menu 2” * “Que é a opção Listar Produtos” |
| Caso digite 3 deverá aparecer:   * “Você escolheu o menu 3” * “Que é a opção Sair do Sistema” | Se digitar qualquer outra coisa:   * “Item de menu inválido” |

Siga os passos para escrever o programa utilizando o switch-case:

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.
2. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_10\_Swich\_case
3. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por ex.: Menu e insira o código mostrado na Figura 3.

|  |
| --- |
| 1. **import** java.util.Scanner; // Import necessário para utilizar a classe Scanner 3. **public** **class** Menu { 4. **public** **static** **void** main ( String [] args ) { 6. System.out.println("======= Menu de Opcoes ========"); 7. System.out.println(" 1 - Cadastrar Produtos "); 8. System.out.println(" 2 - Listas Produtos "); 9. System.out.println(" 3 - Sair do Sistema "); 10. System.out.println("======= Escolha uma opcao ========"); 12. Scanner entrada = **new** Scanner(System.in); 13. **int** menu = entrada.nextInt(); // Lê a opção digitada 15. **switch** (menu) { 16. **case** 1: 17. System.out.println("Voce escolheu o menu 1"); 18. System.out.println("Que eh a opcao Cadastrar Produtos"); 19. **break**; 20. **case** 2: 21. System.out.println("Voce escolheu o menu 2"); 22. System.out.println("Que eh a opcao Listar Produtos"); 23. **break**; 24. **case** 3: 25. System.out.println("Voce escolheu o menu 3"); 26. System.out.println("Que eh a opcao sair do Sistema"); 27. **break**; 28. **default**: 29. System.out.println("Item de menu invalido"); 30. } 31. entrada.close(); 32. } // fim do método main 33. } // fim da classe Menu |

Figura . Código utilizando o switch-case.

1. O resultado do código pode ser visto ao compilar e executar o código, clicando com o botão direito na classe e selecionando a opção Run.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

Na linha 15, o switch recebe a variável menu, que contém o valor da opção escolhida pelo usuário. Caso o usuário digitar 1, o switch testa o case 1 e verifica que ele casa com a opção escolhida pelo usuário, então o bloco de instruções do case 1 será executado:

|  |
| --- |
| case 1:  System.*out*.println("Voce escolheu o menu 1");  System.*out*.println("Que eh a opcao Cadastrar Produtos");  break; |

Graphical user interface, text, application

Description automatically generated

E quando o programa encontra o a instrução break na linha 19, ele encerra o switch e não testa as outras opções.

Caso o usuário digitar 2, o switch testa o case 1, como ele não casa com a opção escolhida, o switch testa o case 2 e verifica que ele casa com a opção escolhida pelo usuário, então o bloco de instruções do case 2 será executado:

|  |
| --- |
| case 2:  System.*out*.println("Voce escolheu o menu 2");  System.*out*.println("Que eh a opcao Listar Produtos");  break; |

Graphical user interface, text, application

Description automatically generated

E quando o programa encontra o a instrução break na linha 23, ele encerra o switch e não testa as outras opções.

Caso o usuário digitar 3, o switch testa o case 1, como ele não casa com a opção escolhida, o switch testa o case 2, como ele não casa com a opção escolhida, o switch testa o case 3 e verifica que ele casa com a opção escolhida pelo usuário, então o bloco de instruções do case 3 será executado:

|  |
| --- |
| case 3:  System.*out*.println("Voce escolheu o menu 3");  System.*out*.println("Que eh a opcao sair do Sistema");  break; |

Graphical user interface, text, application

Description automatically generated

E quando o programa encontra o a instrução break na linha 27, ele encerra o switch e não testa as outras opções.

Se o usuário digitar um valor diferente de 1, 2 ou 3, o switch irá executar a lista de comando do **default**.

|  |
| --- |
| default:  System.*out*.println("Item de menu invalido"); |

Graphical user interface, text, application

Description automatically generated

### Vamos praticar II

Vamos fazer outro exemplo para o switch verificar um char digitado pelo usuário. Siga os passos para escrever o programa utilizando o swtich-case:

1. Crie um novo projeto no InterlliJ IDEA. Para isso, acesse o Menu **File**, a opção **New** e depois clique na opção **Project**.
2. As janelas seguintes são as mesmas descritas na criação de um projeto, portanto se tiver alguma dúvida consulte o material didático. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Projeto\_11\_swich\_case\_II
3. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por ex.: Calculadora e insira o código mostrado na Figura 4.

|  |
| --- |
| 1. **import** java.util.Scanner; // Import necessário para utilizar a classe Scanner 3. **public** **class** Calculadora { 4. **public** **static** **void** main ( String [] args ) { 6. System.out.println("======= Calculadora ========"); 8. Scanner entrada = **new** Scanner(System.in); 9. System.out.println("Digite um número:"); 10. **double** num1 = entrada.nextDouble(); // Lê a opção digitada 11. System.out.println("Digite outro número:"); 12. **double** num2 = entrada.nextDouble(); // Lê a opção digitada 13. System.out.println("======= Operação Matemática ========"); 14. System.out.println(" + Adicionar os números "); 15. System.out.println(" - Subtrair os números "); 16. System.out.println(" \* multiplicar os números "); 17. System.out.println("======== Escolha uma opcao ========="); 18. String oper = entrada.next(); // Lê a opção digitada 20. **switch** (oper) { 21. **case** "+": 22. **double** soma = num1 + num2; 23. System.out.println("Voce escolheu adição"); 24. System.out.println("O número " + num1 + " + " + num2 + " = " + soma); 25. **break**; 26. **case** "-": 27. **double** sub = num1 - num2; 28. System.out.println("Voce escolheu subtração"); 29. System.out.println("O número " + num1 + " - " + num2 + " = " + sub); 30. **break**; 31. **case** "\*": 32. **double** mult = num1 \* num2; 33. System.out.println("Voce escolheu subtração"); 34. System.out.println("O número " + num1 + " \* " + num2 + " = " + mult); 35. **break**; 36. **default**: 37. System.out.println("Operação não disponível ou inválida!"); 38. } 39. entrada.close(); 40. } // fim do método main 41. } // fim da classe Calculadora |

Figura 4. Código utilizando o switch-case.

1. O resultado do código pode ser visto ao compilar e executar o código, clicando com o botão direito na classe e selecionando a opção Run.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

## Exercícios de revisão

1. Ler dois valores inteiros:

* e se forem iguais, mostrar “Números iguais”;
* caso contrário, apresentar a diferença do maior pelo menor.

1. Ler dois números (ponto ‑utuante) e apresentá-los em ordem decrescente.

* supor que não sejam iguais.

1. Faça um algoritmo para ler um número inteiro.
   * verifique se o número está no intervalo entre 50 (inclusive) e 100 (inclusive);
   * se estiver:

* imprimir “Pertence ao intervalo”;
  + senão:
* imprimir “Não pertence ao intervalo”

1. Faça um algoritmo que leia a altura e o sexo de uma pessoa:
   * calcule e mostre seu peso ideal.
   * usar as fórmulas a seguir para calcular o peso ideal:

* para o sexo masculino: (72,7 \* altura) – 58
* para o sexo feminino: (62,1 \* altura) – 44,7

1. O IOS concedeu um bônus de 20% do valor do salário a todos os funcionários com tempos de trabalho na empresa igual ou superior a cinco anos e de 10% aos demais funcionários. Faça um algoritmo que leia o salário e a quantidade de anos de cada funcionário, calcule e imprima o valor do bônus.
2. Faça um algoritmo que verifique a validade de uma senha fornecida pelo usuário.
   * sabendo que a senha é R10p5:

* imprimir mensagem de “acesso concedido” ou “acesso negado”.
  + Para comparar duas Strings utilizar o método equals();
* ele retorna um valor booleano!
* Exemplo:
  + if(senha.equals("R10p5"))

1. A prefeitura do Rio de Janeiro abriu uma linha de crédito para os funcionários estatuários.
   * o valor máximo da prestação não poderá ultrapassar 30% do salário bruto.
   * fazer um algoritmo que leia o salário bruto e o valor da prestação e informar se o empréstimo pode ou não ser concedido.
   * Exemplo:

* Salário bruto: 1200,00
* Valor da prestação: 400,00
* Empréstimo não pode ser concedido!

1. Faça um algoritmo para encontrar o maior número entre 3 números inteiros.
   * O algoritmo deve ler três inteiros;
   * se forem todos iguais, imprimir: “os números são iguais”;
   * caso contrário, imprimir o maior dos 3 números.
2. Ler o salário de uma pessoa e calcular e imprimir o desconto do INSS (calculado!!) de acordo com a tabela a seguir:

Text

Description automatically generated

1. Criar uma calculadora de operações básicas:
   * soma, subtração, multiplicação e divisão.
   * o algoritmo deve ler dois números e o sinal correspondente à operação desejada;
   * utilize o tipo char para ler a operação;
   * no final deve ser impresso o resultado.
   * Restrições:

* se o sinal digitado não corresponder a uma operação. . .
  + apresentar a mensagem Sinal Inválido e finalizar.
* para a operação de divisão verificar se o divisor é válido (maior que zero)!
  + caso seja menor ou igual a zero, informar a mensagem “Impossível dividir!!”

1. Crie um algoritmo que leia um número de 1 a 7.
   * imprima o dia da semana correspondente;
   * assuma que 1 corresponde ao Domingo.
2. Faça um algoritmo que receba dois números;
   * execute as operações listadas a seguir . . .
   * de acordo com a escolha do usuário.

A picture containing table

Description automatically generated

1. Faça um algoritmo que leia um inteiro entre 1 e 12.
   * imprima o nome do mês por extenso.
2. Um funcionário receberá aumento de acordo com seu plano de trabalho.
   * Faça um algoritmo que leia:

* o plano de trabalho;
* e o salário atual de um funcionário;
  + Calcule e imprima o seu novo salário.

Table

Description automatically generated with low confidence

1. Dado o cardápio de uma lanchonete:

Table

Description automatically generated

Faça um algoritmo que:

* leia o código do produto e a quantidade;
* calcule o valor a ser pago pelo cliente;
* imprimir o valor a ser pago;
* imprimir o nome do produto.