# Aula 03-Desvios condicionais – if/else

## 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 if/else.

## Estrutura de um programa em Java

O Java é uma linguagem orientada a objeto, isto é, todo o código de um programa deve estar definido dentro de uma classe. Além disso, o código deve estar definido dentro de um método especial chamado **main()**. O método **main()** é o ponto de partida na execução de um programa em Java.

Graphical user interface, text, application

Description automatically generated

No início do aprendizado de uma linguagem de programação, geralmente, o resultado do processamento de um algoritmo pode ser armazenado em uma variável e/ou pode ser exibido diretamente em um dispositivo de saída. O monitor (console) é o dispositivo de saída padrão de um projeto em Java. Como foi visto o comando de saída utilizado para escrever dados e resultados na saída padrão (monitor/console) é:

System.out.println("");

Por exemplo, o código abaixo:

A picture containing text

Description automatically generated

Produz o seguinte resultado mostrado na tela do computador.

Graphical user interface, text

Description automatically generated

## Comando de entrada

Além de imprimir os dados e resultado na saída padrão do computador, podemos também ler dados digitado de um teclado e armazená-los na memória do computador por meio de variáveis. Para isso, é necessário criar um buffer para guardar os dados digitados no teclado de depois transferi-los para a memória (variável).

A picture containing text, clock

Description automatically generated

Para criar o buffer é necessário utilizar a classe Scanner:

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

Description automatically generated

Após criar o buffer do teclado basta transferir esse dado para a variável. A leitura de um dado é realizada associando o tipo de entrada ao tipo da variável que receberá o dado.

Table

Description automatically generated with low confidence

Para usar a classe Scanner é necessário fazer um import de um pacote antes da definição da classe.

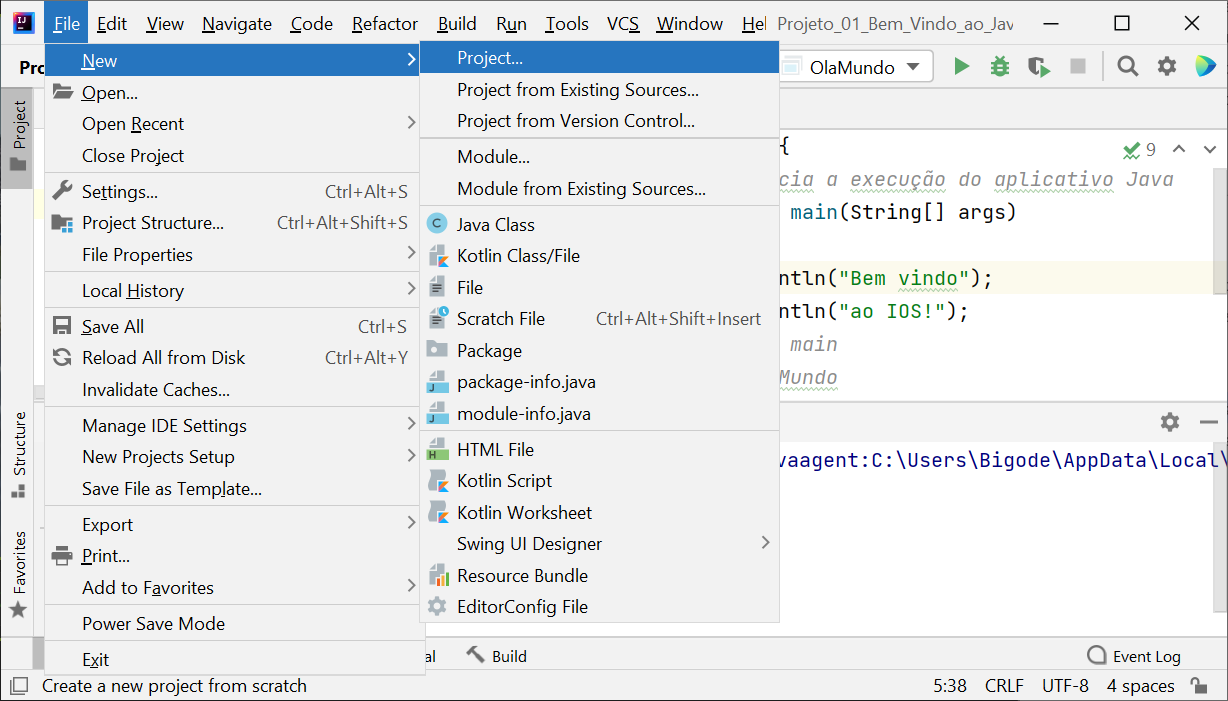
A picture containing text

Description automatically generated

### Vamos programar

Vamos fazer um programa em que solicitaremos para o usuário que informe o nome e o valor de duas notas de um aluno. O programa deverá calcular a média dessas notas e depois imprimir a média calculada. Nesse exemplo, precisaremos de três variáveis duas para armazenar as notas digitadas e outra para armazenar a média calculada. Portanto, siga os passos para escrever o nosso programa para calcular a média das notas:

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\_05\_Comando\_Entrada
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: Media e insira o código mostrado na Figura 1.

|  |
| --- |
| 1. **import** java.util.Scanner; // Import necessário para utilizar a classe 3. **public** **class** Media { 4. **public** **static** **void** main (String [] args) { 5. // Intância do objeto (buffer) utilizando a classe Scanner 6. Scanner entrada = **new** Scanner(System.in); 7. // Declaração de variáveis 8. String aluno; 9. **double** nota1 , nota2 , media; 10. System.out.println ("Digite seu nome: "); 11. aluno = entrada.nextLine();    // Recebe o nome do aluno 13. System.out.println ("Digite duas notas"); 14. nota1 = entrada.nextDouble();  // Recebe a primeira nota 15. nota2 = entrada.nextDouble();  // Recebe a segunda nota 17. media = ( nota1 + nota2 ) / 2; // Calcula a média das notas 19. System.out.println(aluno + " sua média é: " + media); // Exibe a média 20. entrada.close();  // Encerra a instância 21. } // fim do método main 22. } // fim da classe Media |

Figura 1 - Código do programa para ler dados digitados no teclado.

1. Ao executar o programa, ele irá solicitar o nome do aluno e duas notas.

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

Description automatically generated

1. E o resultado do cálculo da média é exibido em seguida.

A picture containing text

Description automatically generated

A instrução na linha 6 da Figura 1 cria uma instancia da classe Scanner, que irá receber os dados digitados no teclado.

Na linha 11, a instrução lê o nome e armazena na variável aluno. Como o dado esperado é uma string, então utilizamos **entrada.nextLine()** para ler o valor como uma string.

Na linha 14, a instrução lê a primeira nota e armazena na variável nota1. Como o dado esperado é um double (valor numérico com casas decimais), então utilizamos **entrada.nextDouble()** para ler o valor como um double.

Na linha 15, a instrução recebe a segunda nota e armazena na variável nota2. Como o dado esperado é um double (valor numérico com casas decimais), então utilizamos **entrada.nextDouble()** para ler o valor como um double.

Na linha 17, a instrução **media = ( nota1 + nota2 ) / 2** calcula a média das notas recebidas e o resultado é impresso através do comando na 19.

Por fim, na linha 20, o método close() encerrar a instância da classe.

## Estruturas condicionais

Geralmente, as instruções de um programa são executadas sequencialmente uma após a outra, na ordem em que foram escritas. Porém muitas vezes é necessário alterar a sequência da execução de um programa de acordo com o resultado de um teste lógico ou até mesmo repetir um determinado trecho de código por inúmeras vezes. Nesse contexto, as estruturas condicionais permitem decidir o fluxo de execução de programa, selecionando as instruções que serão executadas de acordo com um teste lógico (Desvio condicional).

O Desvio Condicional é usado quando existe a necessidade de verificar condições para executar uma instrução ou um bloco de instruções. Toda condição testada pode retornar dois valores lógicos: true ou false. Vejamos alguns exemplos de condições:

* (x > y)
* (peso < 50.0)
* ((x > 0) && (x < 8))
* ((x == 5 && y == 2)|| y == 3)

O desvio condicional pode ser: simples (if), composto (if/else) ou encadeados (aninhados).

### Desvio condicional simples (if)

O desvio condicional simples é usado para verificar se uma dada condição é atendida. Caso a condição seja atendida um determinado conjunto de instruções deverá ser executado. Caso contrário, a condição não for atendida, o fluxo de execução do algoritmo seguirá após o fim do bloco de decisão.

Diagram

Description automatically generated

A sintaxe do comando if é:

Text

Description automatically generated with low confidence

Fique atento as chaves de início e fim do bloco de instruções do desvio condicional if.

Atenção: se o bloco de instruções do if tiver apenas uma instrução, o uso de abertura e fechamento de chaves é opcional.

### Vamos praticar

Vamos criar um programa para verificar o número de pontos de um jogador. Caso o número de pontos for maior ou igual a 1000, o programa deverá imprimir uma mensagem que o jogador passou para a segunda fase.

Diagram

Description automatically generated

Siga os passos para escrever o programa utilizando o desvio condicional simples:

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\_06\_Devio\_Cond\_if
3. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por ex.: Jogo e insira o código mostrado na Figura 2.

|  |
| --- |
| 1. **import** java.util.Scanner; // Import necessário para utilizar a classe 3. **public** **class** Jogo { 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** pontos = 0; 8. System.out.println ("Digite o número de pontos do jogador: "); 9. pontos = entrada.nextInt();    // Recebe o número de pontos 11. **if** (pontos >= 1000) { 12. System.out.println ("Você passou para a fase 2!!!"); 13. } 15. System.out.println("proxima instrucao"); 16. entrada.close(); 17. } // fim do método main 18. } // fim da classe Jogo |

Figura 2. Código utilizando o desvio condicional simples.

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

Description automatically generated

Figura 4. Resultado da execução do código.

Na linha 11, o comando if testa se o valor digitado pelo usuário é maior ou igual a 1000. Caso a condição seja verdadeira, a mensagem da segunda fase é mostrada (Figura 4). Caso contrário, a condição seja falsa, apenas a mensagem de próxima instrução é impressa.

Graphical user interface, text, application

Description automatically generated

Nesse exemplo, o desvio condicional if tem apenas uma instrução, portanto o uso de chaves é opcional:

|  |
| --- |
| if (pontos >= 1000)  System.*out*.println ("Você passou para a fase 2!!!"); |

### Desvio condicional composto (if/else)

O desvio condicional composto prevê dois conjuntos de instruções para serem executados de acordo com a avaliação da condição. Nesse caso, esse um conjunto de instruções para serem executados se a condição for verdadeira e um outro conjunto de instruções para serem executados se a condição for falsa.

Diagram

Description automatically generated

O comando if/else possui a seguinte sintaxe:

Graphical user interface, text

Description automatically generated

Nesse caso, temos uma abertura e fechamento de chaves delimitando o bloco de instruções do if, que serão executadas caso a condição testada seja verdadeira, e outra abertura e fechamento de chaves delimitando o bloco de instruções do else, que serão executadas caso a condição testada seja falsa.

### Vamos praticar

Vamos criar um programa para verificar se um número digitado pelo teclado é par ou ímpar. Caso o número seja par, o programa deverá imprimir a mensagem que ele é par e, caso o número seja ímpar, o programa deverá imprimir a mensagem que ele é ímpar.

Diagram

Description automatically generated

Siga os passos para escrever o programa utilizando o desvio condicional composto:

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\_07\_Devio\_Cond\_if\_else
3. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por ex.: ParImpar e insira o código mostrado na Figura 4.

|  |
| --- |
| 1. **import** java.util.Scanner; // Import necessário para utilizar a classe 3. **public** **class** ParImpar { 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** numero; 8. System.out.println ("Digite um número inteiro: "); 9. numero = entrada.nextInt();    // Recebe o número 11. // Verifica se o número é par 12. **if** ((numero % 2) == 0) { 13. System.out.println ("O número " + numero + " é par"); 14. } 15. **else** { 16. System.out.println ("O número " + numero + " é ímpar"); 17. } 18. entrada.close(); 19. } // fim do método main 20. } // fim da classe ParImpar |

Figura 4. Código utilizando o desvio condicional composto.

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 generated Graphical user interface, text, application

Description automatically generated

Figura 5. Resultado da execução do código.

Na linha 12, o comando if testa se o resto da divisão do valor digitado por 2 é igual a 0. Caso a condição seja verdadeira, o número é par, então o bloco de instruções do comando if deverá ser executado e a mensagem de número par deverá ser impressa. Caso contrário, a condição seja falsa, o número é ímpar, então o bloco de instruções do comando else deverá ser executado e a mensagem de número ímpar deverá ser impressa.

Nesse exemplo, o desvio condicional if/else tem apenas uma instrução, portanto o uso de chaves é opcional:

|  |
| --- |
| if ((numero % 2) == 0)  System.*out*.println ("O número " + numero + " é par"); else   System.*out*.println ("O número " + numero + " é ímpar"); |

Você também poderia usar o operador lógico de negação na condição testada e desse modo não precisaria colocar o operador relacional de comparação ==.

|  |
| --- |
| if (!(numero % 2))  System.*out*.println ("O número " + numero + " é par"); else   System.*out*.println ("O número " + numero + " é ímpar"); |

O resultado da operação **numero % 2** só poderá ser zero (número par) ou um (número ímpar). Como queremos que seja verdadeira a condição quando o número for par, então colocamos o **operador lógico de negação !** para inverter o valor booleano. Portanto, se o resultado for **zero**, o operador inverte para **um** e assim é verdadeira a condição se o número é par. Se o resultado for **um**, o operador inverte para **zero** e assim é falsa a condição se o número é ímpar.

## Exercícios de revisão

1. Faça um algoritmo que leia um número inteiro.

* Se o número for maior que 20. . .
* calcular e imprimir a metade dele.

1. Faça um algoritmo que:

* receba a idade de uma pessoa;
* se for maior de idade imprima: “maior de idade”;
* senão imprima: “menor de idade”.

1. Ler dois números (double) e apresentá-los em ordem decrescente. Supor que não sejam iguais.
2. Faça um algoritmo que receba a idade de uma pessoa e mostre se ela é maior de idade ou não.