# Aula 07-Teste de Mesa

## 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 um teste de mesa.

## Teste de mesa

O teste de mesa é uma análise para descobrir se um programa funciona logicamente. Basicamente, o Teste de Mesa (Trace Table) é um processo manual que é utilizado para validar a lógica de um determinado algoritmo. Esse processo era realizado com papel e caneta em cima de uma mesa e o programador ia seguindo o fluxo do algoritmo e atualizando os valores das variáveis do programa manualmente a cada instrução do algoritmo que era analisada.

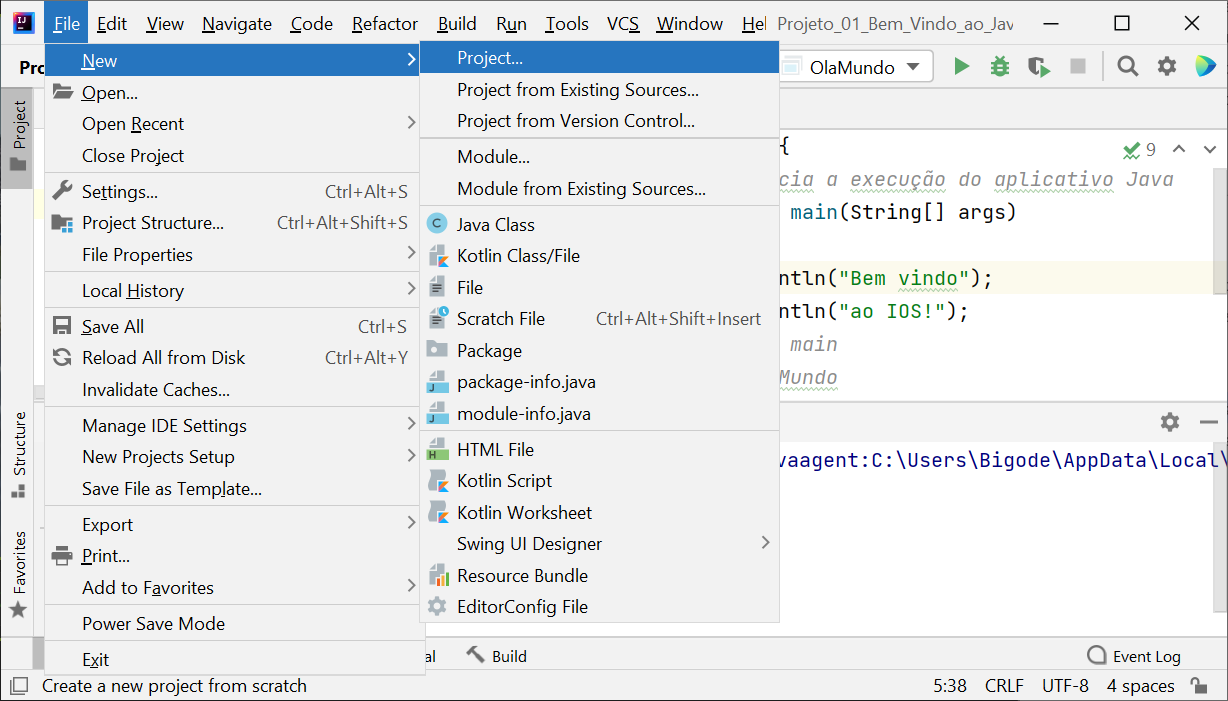
Esse teste é utilizado principalmente em algoritmos quando a linguagem utilizada não possui nenhuma ferramenta automatizada de depuração. Como as linguagens de programação atuais costumam possuir tais ferramentas, é mais comum utilizá-las a fazer o teste de mesa propriamente dito, embora para quem ainda é iniciante, eu particularmente ainda recomendo utilizá-lo, visto que provavelmente não terá domínio sobre a ferramenta de depuração.

Então essa parte da disciplina de programação em Java pode ser vista mais como um desenvolvimento de habilidades de programação para visualizar como as variáveis são atualizadas a cada iteração de um loop, por exemplo. Não vamos ter que fazer teste de mesa manualmente.

### Vamos praticar

Vamos fazer um programa que tenha uma variável de controle, uma variável acumuladora e a condição testada de um laço de repetição. Siga os passos para criar o projeto:

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\_22\_teste\_de\_mesa
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: **Teste** e insira o código mostrado na Figura 1.
3. **public** **class** Teste {
4. **public** **static** **void** main(String[] args) {
5. **int** controle, acumulador = 0;
6. **boolean** condicao;
7. System.out.println("Numero da iteração\tCondição\tControle\tAcumulador");
8. **for**(controle = 1; controle <= 10; controle++){
9. acumulador += controle;
10. condicao = controle <= 10;
11. System.out.println("Iteração "+ controle+ "\t\t\t"+ condicao + "\t\t\t" + controle + "\t\t\t" + acumulador);
12. }
13. } // fim do método main
14. }

Figura 1 - Código do programa teste de mesa.

1. Ao executar o programa, você consegue visualizar o valor de cada variável em cada iteração do laço de repetição.

Table

Description automatically generated

Observe que quando a variável de controle assumiu o valor 11, a condição testada se tornou falsa e então as instruções dentro do for não foram executadas mais. Por isso, paramos de imprimir na iteração 10 e mostrando que nesse momento a condição ainda era verdadeira.

### Vamos praticar II

Vamos fazer um programa de teste de mesa para o comando while. Siga os passos para criar o projeto:

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\_23\_Teste\_de\_mesa
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: **Teste2** e insira o código mostrado na Figura 2.
3. **public** **class** Teste2 {
4. **public** **static** **void** main(String[] args) {
5. **int** controle = 10;
6. **long** acumulador = 1;
7. **boolean** condicao;
8. System.out.println("Numero da iteração\tCondição\tControle\tAcumulador");
9. **while**(controle > 0){
10. **int** iteracao = 10 - controle + 1;
11. acumulador \*= controle;
12. condicao = controle > 0;
13. System.out.println("Iteração "+ iteracao+ "\t\t\t"+ condicao + "\t\t\t" + controle + "\t\t\t" + acumulador);
14. controle--;
15. }
16. } // fim do método main
17. }

Figura 2 - Código do programa do teste de mesa com o while.

1. Ao executar o programa, você consegue visualizar o valor de cada variável em cada iteração do laço de repetição.

Table

Description automatically generated

### Vamos praticar III

Vamos fazer um programa de teste de mesa para o comando do-while. Siga os passos para criar o projeto:

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\_24\_teste\_de\_mesa
2. No diretório **src**, crie um arquivo para a classe. Não se esqueça de dar um nome representativo para o projeto, por exemplo: **Teste3** e insira o código mostrado na Figura 3.
3. **public** **class** Teste3 {
4. **public** **static** **void** main(String[] args) {
5. **int** controle = 1;
6. **long** acumulador = 100;
7. **boolean** condicao;
8. System.out.println("Numero da iteração\tCondição\tControle\tAcumulador");
9. **do**{
10. acumulador -= controle;
11. condicao = controle <=10;
12. System.out.println("Iteração "+ controle+ "\t\t\t"+ condicao + "\t\t\t" + controle + "\t\t\t" + acumulador);
13. controle++;
14. }**while**(controle <=10);
15. } // fim do método main
16. }

Figura 3 - Código do programa teste de mesa do-while.

1. Ao executar o programa, ele irá imprimir os valores das variáveis de controle do **for** e do **while**.

Table

Description automatically generated

## Ferramentas de debug do IntelliJ

O teste de mesa é interessante quando você não tiver as ferramentas de debug em uma IDE, mas se sua IDE já possui esse tipo de ferramenta, ele se torna apenas uma brincadeira para testarmos instruções e formatar saídas. Isso porque, é muito mais útil você aprender a utilizar as ferramentas de depuração (debug) disponíveis na IDE que você está utilizando, do que inserir instruções no código que não são necessárias para o funcionamento principal do programa.

Na IDE IntelliJ, você precisa marcar um ou mais breakpoint (ponto de parrada). VocÊ pode fazer isso clicando duas vezes ao lado do número da linha desejada. Por exemplo, no Projeto\_22\_teste\_de\_mesa. Vamos inserir um breakpoint na linha 6, onde está o comando for:

Graphical user interface, text, application, email

Description automatically generated

Agora, você pode acessar a ferramenta de debug no Menu Run → Debug

Graphical user interface, text, application, email

Description automatically generated

Observe que ao acessar o Debug do programa, as variáveis que já foram inicializadas com algum valor já aparecem na janela do depurador. Se você clicar uma vez no botão Resume Program (tecla de atalho F9), o programa irá executar uma iteração do for e parar na linha novamente.

Graphical user interface, text, application

Description automatically generated

Você pode clicar várias vezes até acabar as iterações do comando for:

Graphical user interface, text, application, email

Description automatically generated

Se clicar mais uma vez, o debug chegará ao final do programa, terminando assim a sua execução.

## Exercícios de revisão

1. Faça um programa que imprima a seguinte sequência: 100, 99, 98, ... 1.
2. Faça um algoritmo que apresente na tela os quadrados dos números inteiros de 15 a 100.
   * Exemplo:
     + O quadrado de 15 é 225
     + O quadrado de 16 é 256
     + ...
     + O quadrado de 100 é 10000
3. Ler um número inteiro e imprimir na tela a sequência de números que vai do número 1 até o número lido:
   * Exemplo:
     + Digite um número:
     + 8
     + Sequência: 1 2 3 4 5 6 7 8..
4. Faça um algoritmo que receba a idade e a altura de 10 pessoas:
   * calcule e mostre a média das alturas daquelas com mais de 50 anos.
5. Faça um algoritmo que receba duas notas de 6 alunos, calcule e mostre:
   * A média aritmética das duas notas de cada aluno;
   * Forneça a seguinte mensagem de acordo com a nota:
     + **REPROVADO** se média é menor ou igual a 3.
     + **EXAME** se média é acima de 3 e menor que 7.
     + **APROVADO** maior ou igual a 7
   * O total de alunos aprovados.
   * O total de alunos de exame.
   * O total de alunos reprovados.
   * A média da classe.
6. Faça um algoritmo que receba a idade, a altura e o peso de 10 pessoas, calcule e mostre:
   * A quantidade de pessoas maiores de 50 anos.
   * A média das alturas das pessoas com idade entre 10 e 20 anos.
   * A porcentagem de pessoas com peso inferior a 40 quilos.