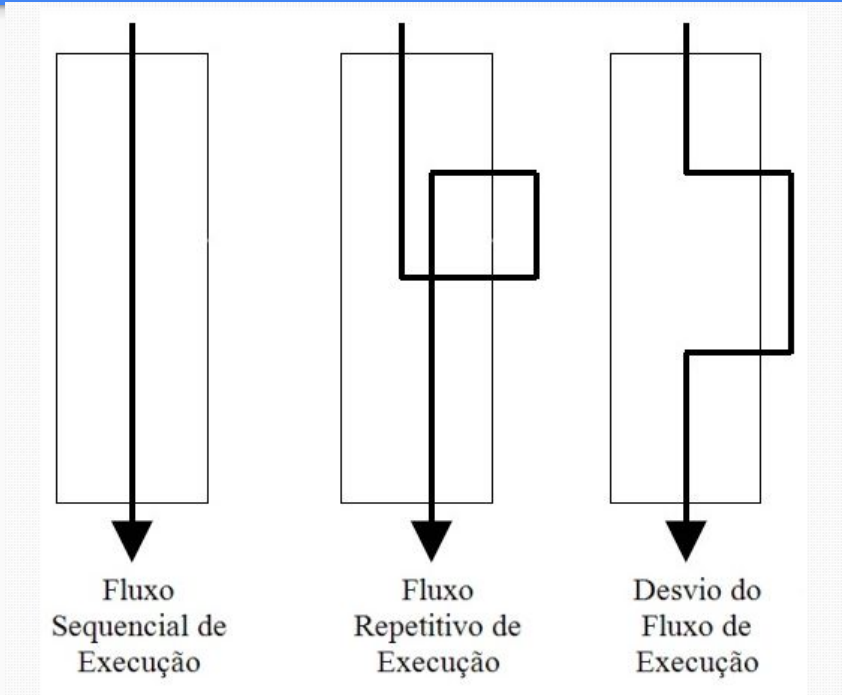
PROGRAMAÇÃO I

AULA 04 - Prof. Maycon de Moraes

Tópicos da Aula

- Revisão da validação do Eclipse
- Aprofundando os conhecimentos no Java
  - Controle de Decisão
  - Controle de Repetição
  - IO de dados
- Exercícios de Fixação

Tipos de Fluxos



Estrutura Condicional

```
1 | if (expressão booleana) {
2 |     // bloco de código 1
3 | } else {
4 |     // bloco de código 2
5 | }
```

A estrutura condicional if/else permite ao programa avaliar uma expressão como sendo verdadeira ou falsa e, de acordo com o resultado dessa verificação, executar uma ou outra rotina.

Na linguagem Java o tipo resultante dessa expressão deve ser sempre um boolean, pois diferentemente das demais, o Java não converte null ou inteiros como 0 e 1 para os valores true ou false

Estrutura Condicional

```
1 if (expressão booleana 1) {
2     // bloco de código 1
3 } else if (expressão booleana 2) {
4     // bloco de código 2
5 } else {
6     // bloco de código 3
7 }
```

Complementar ao if/else temos o operador else if. Esse recurso possibilita adicionar uma nova condição à estrutura de decisão para atender a lógica sendo implementada.

Podemos utilizar quantos else if forem necessários. Entretanto, o else deve ser adicionado apenas uma vez, como alternativa ao caso de todos os testes terem falhado

Operador Ternário

```
1 | (expressão booleana) ? código 1 : código 2;
```

O operador ternário é um recurso para tomada de decisões com objetivo similar ao do if/else, mas que é codificado em apenas uma linha.

Ao avaliar a expressão booleana, caso ela seja verdadeira, o código 1, declarado após o ponto de interrogação (?) será executado; do contrário, o programa irá executar o código 2, declarado após os dois pontos (:).

While

```
int x = 15;

while (x < 18) {
    System.out.println("Você não tem permissão para entrar");
    x++;
}
```

O termo while pode ser traduzido para o português como “enquanto”. Este termo é utilizado para construir uma estrutura de repetição que executa, repetidamente, uma única instrução ou um bloco delas “enquanto” uma expressão booleana for verdadeira.

Do-While

```
int x = 15;

do {

    System.out.println("Você não tem permissão para entrar");
    x++;

} while (x < 18);
```

Em um laço while, a condição é testada antes da primeira execução das instruções que compõem seu corpo Em um laço do-while, por outro lado, a condição somente é avaliada depois que suas instruções são executadas pela primeira vez, assim, mesmo que a condição desse laço seja falsa antes de ele iniciar, suas instruções serão executadas pelo menos uma vez.

For

```
for(int x = 0; x<1000; x++){
    System.out.println("Valor: " + x);
}
```

O laço for é uma estrutura de repetição compacta. Seus elementos de inicialização, condição e iteração são reunidos na forma de um cabeçalho e o corpo é disposto em seguida.

O laço for e o laço while são apenas formas diferentes de uma mesma estrutura básica de repetição. Qualquer laço for pode ser transcrito em termos de um laço while e vice-versa.

For “Melhorado”

```
int[] array = {1,2,3,4,5};

for (int i : array) {
    System.out.println(i);
}
```

O enhanced-for foi introduzido a partir do Java 5, e é utilizado para realizar as varreduras em collections. Para cada iteração do for, o elemento da iteração é atribuído à variável. Utilizando o enhanced-for, você é obrigado a percorrer um array por exemplo.

Quebras de Laço

As quebras de laço são utilizadas para interromper o fluxo normal das estruturas de repetição while, do-while e for. Há dois tipos distintos de quebras de laço, representadas pelas palavras reservadas break e continue.

- Break
- Continue

Entrada de Dados: Classe Scanner

```
Scanner scanner = new Scanner(System.in);
int numeroInteiro = scanner.nextInt();
byte numeroByte = scanner.nextByte();
long numeroLong = scanner.nextLong();
boolean booleano = scanner.nextBoolean();
float numeroFloat = scanner.nextFloat();
double numeroDouble = scanner.nextDouble();
```

A classe Scanner apareceu a partir do Java 5 com o objetivo de facilitar a entrada de dados no modo Console.

Uma das características mais interessante da classe Scanner é a possibilidade de obter o valor digitado diretamente no formato do tipo primitivo que o usuário digitar.

Atividade 01

- Refazer a calculadora da aula 02 com números digitados pelo usuário

Atividade 02

Em uma competição de salto em distância cada atleta tem direito a cinco saltos. O resultado do atleta será determinado pela média dos cinco valores restantes. Você deve fazer um programa que receba o nome e as cinco distâncias alcançadas pelo atleta em seus saltos e depois informe o nome, os saltos e a média dos saltos. O programa deve ser encerrado quando não for informado o nome do atleta. A saída do programa deve ser conforme o exemplo ao lado:

```
Atleta: Rodrigo Curvêllo

Primeiro Salto: 6.5 m
Segundo Salto: 6.1 m
Terceiro Salto: 6.2 m
Quarto Salto: 5.4 m
Quinto Salto: 5.3 m

Resultado final:
Atleta: Rodrigo Curvêllo
Saltos: 6.5 - 6.1 - 6.2 - 5.4 - 5.3
Média dos saltos: 5.9 m
```

Atividade 03

Em uma eleição presidencial existem quatro candidatos. Os votos são informados por meio de código. Os códigos utilizados são:

```
1 , 2, 3, 4  - Votos para os respectivos candidatos
(você deve montar a tabela ex: 1 - Jose/ 2- João/etc)
5 - Voto Nulo
6 - Voto em Branco
```

Faça um programa que calcule e mostre:

- O total de votos para cada candidato;
- O total de votos nulos;
- O total de votos em branco;
- A percentagem de votos nulos sobre o total de votos;
- A percentagem de votos em branco sobre o total de votos. Para finalizar o conjunto de votos tem-se o valor zero.