AULA 4
Comandos - Caixa de Texto

Utilizando da Instrução Scanner

```
public class Soma2 {
        public static void main(String arg []){
            Scanner entrada = new Scanner(System.in);


     int numero ,numero2, soma;


            System.out.println ("Digite um numero ");
            numero = entrada.nextInt();


            System.out.println("Digite o segundo numero");
            numero2 = entrada.nextInt();


            soma = numero + numero2;
            System.out.println("A soma foi " + soma);
        }
}
```

Inserir o texto em uma caixa de Diálogo

- Utilizando da Instrução Scanner;

  Scanner entrada = new Scanner (System.in)

- É uma instrução de declaração de variável que especifica o tipo da variavel (entrada) com um Scanner.

- Um Scanner permite a um programa ler os dados , por exemplo digitados pelo usuário.
  - Numero = entrada.nextInt();

- o método nextInt() obtém o valor digitado pelo usuário, no momento em que o usuário clica em ok

Programinha

Exercício 1

- Utilizando a classe Scanner faça o programa que calcule a media dos usuário e diga se o mesmo passou ou perdeu de ano.

Exibir o texto em uma caixa de Diálogo

```
public class Dialogo
{

public static void main(String args[ ] ) {

JOptionPane.showMessageDialog(null, "Bem Vindo ao Mundo Java");

    }
}
```

Inserir o texto em uma caixa de Diálogo

```
public class NomeDialogo
{

public static void main(String args[ ] ) {

String nome;
            nome = JOptionPane.showInputDialog("Qual o seu nome?");
            System.out.println(nome);
    }
}
```

AULA 5
Comandos – Repetição

Os comandos de Repetição

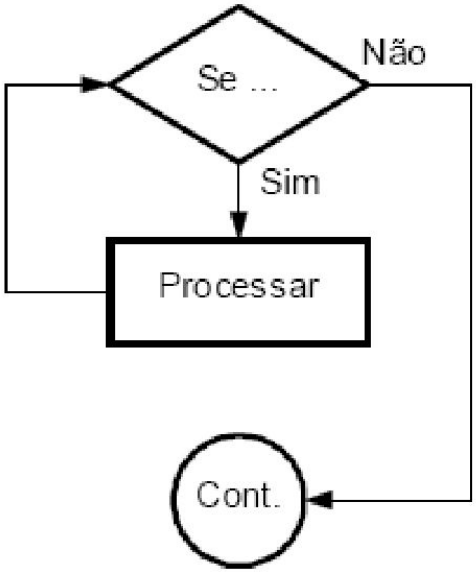
- São comandos que permitem que um determinado conjunto de instruções ou comandos sejam executados um número definido ou indefinido de vezes, ou enquanto um determinado estado de coisas prevalecer ou até que seja alcançado.

- While
- For

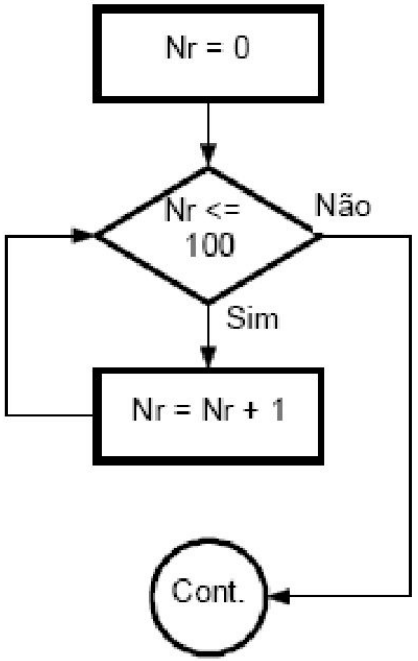
Pseudo-Linguagens

While

- Neste caso, o bloco de operações será executado enquanto a condição x for verdadeira.

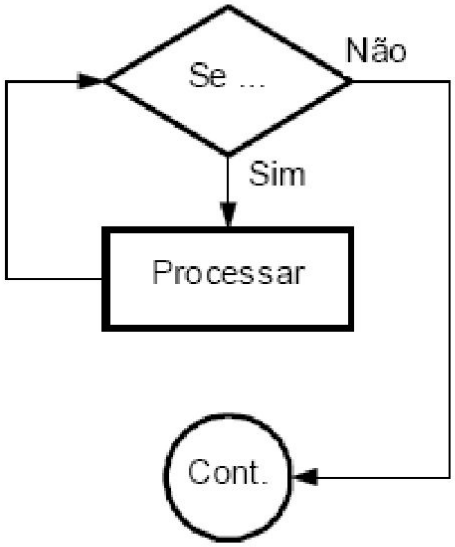


Exemplo de Contador

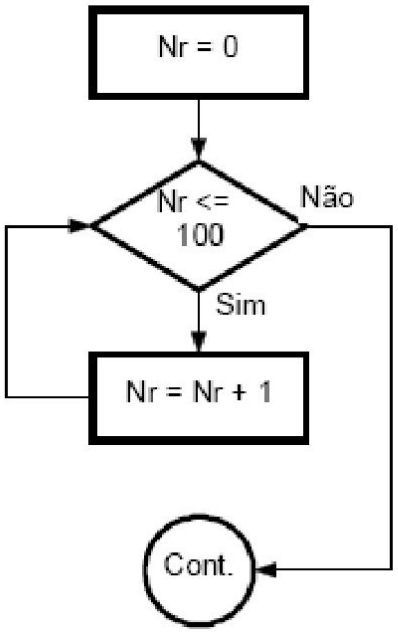


While

Pseudo-Linguagens



Exemplo de Contador



- Pseudocódigo

  Enquanto (Nr <=100) repetir

  Calcula Nr = Nr + 1

While

- While é uma estrutura de repetição.
- While executa uma comparação com a variável.
  - Se a comparação for verdadeira, ele executa o bloco de instruções ({ }) ou apenas a próxima linha de código logo abaixo.
- Estrutra

```
WHILE (comparação)
{
  //Instruções
}
```

While - Programinha

Escreva um contador que conte de 0 e vá até 50.

Utilizando o while.

While - Programinha

```
public class ExemploWhile {

public static void main(String args[]) {

 int contador = 0;
while (contador  < = 50) {
   System.out.println("Repetição nr: " + contador);
   contador++;
        }
   }
}
```

For

- **For** (*o Para ... Até ... Faça?)*
  - Neste caso, o bloco de operações já possui um número inicial e final para uma variável de controle.
  - Estrutura

    for (var=valor inicial; condição; incremento)

    comando
- Exemplo :

  for (contador=3; contador<=11; contador ++)

  Imprima “Contador”;
- Pseudocódigo

  Para contador= 3 até 11 repetir

  Mostrar contador;

For - Programinha

Escreva um contador que conte de 0 e vá até 50. Utilizando o For;

For - Programinha

```
public class Contador {

    public static void main(String args[]){

        int i;
        int limite = 50;

        for (i=0; i<limite; i++){
          System.out.println("Meu Contador " + i);
        }
    }
}
```

Exercício

Calcular o juros e o montante de uma capital de 100,00 investido numa caderneta de poupança que rende 10% ao mês. Durante 1 ano.

```
public class JurosSimples {

public static void main(String arg []){
        int tempo = 0;
        int capital = 100;
        double juros, montante
        double taxa = 0.10;

 for (tempo = 1; tempo <=12; tempo ++){
            juros = capital * taxa * tempo;
            montante = capital + juros;
  System.out.println(tempo + " Mes" + " juros de : = " +
   juros +" Montante = "+ montante );
        }
    }
}
```

Exercício

Quem fez com For faz agora com While e quem fez com While faz agora com For!

Questionamento???

- Para Fazer um contador de Dinheiro para o caixa de supermercado se utiliza o For ou o While?

- Para Fazer uma máquina de Dinheiro que passe o troco para o cliente se utiliza o For ou o While?

Utilizando o Length

- O método length permite descobrir o número de caráteres contidos numa String.

Por exemplo:

```
int a;
nome = JOptionPane.showInputDialog("Digite seu nome");
a = nome.length();
System.out.println("O Tamanho do Nome é":  a);
```

Igualdade entre String

- Utilizando o método *Equals*

```
String nome = “Pedro";
if (nome.equals(“Pedro")) {
System.out.println("Nome é Pedro");
}
else {
System.out.println("Nome não é Pedro");
}
```

Quinto Programa

Exercício

- Pergunte o nome completo do usuário e logo após, imprima na tela:
- Olá **nome_completo** seu nome possui **X** caracteres.

Break

- O comando break serve para determinar uma quebra de estrutura, ou seja, ele faz com que, por exemplo, um loop (repetição) pare.

```
for (int x=1; x<=1000; x++)
{
 System.out.println ("Esta é a repetição nr: "+ x);
      if (x==10)
      break;

}
```

Switch, Case e Default

- Já discutimos a instrução if de uma única seleção e a instrução dupla if...else.

- O JAVA fornece a instrução de múltipla seleção switch para realizar diferentes ações baseadas nos possíveis valores

Switch

- A estrutura switch verifica uma variável e age de acordo com seus cases. Os cases são as possibilidades de resultados que são obtidos por switch.

- A estrutura do Switch é:

```
SWITCH (variável) {
    CASE valor :
    instrução
  }
```

Switch, Case e Default

- **Dica importante**:
  - É um bom costume sempre terminar um código após o case com um comando break. Assim, nós evitamos que o resto do código seja executado por acidente.
  - E vale também ressaltar que case não gera resultados booleanos, portanto, não há a possibilidade de fazer comparações
    - Ex: case var1 > var2:

Switch e Default

- Como switch pode receber várias possibilidades, pode ocorrer de algum caso estar fora do alcance ou não definido.
- Nesse momento, default faz seu papel. Default pega qualquer resultado que não esteja definido no case.
  - Podemos colocá-lo onde quisermos dentro de switch , mas, geralmente, o colocamos no final.

```
public class ExemploSwitch {

public static void main(String args[]) {
int diaDaSemana = 2;

switch (diaDaSemana) {
case 1:
    System.out.println("Domingo");
    break;
case 2:
    System.out.println("Segunda-feira");
    break;
default:
    System.out.println("Este não é um dia válido!");
    break;
    }
    }
 }
```

Criando Métodos

- Exemplo de Uma Função Soma;

JAVA BÁSICO

Introdução à POO – Programação Orientada a Objetos

Introdução Programação Orientada a Objetos

- A **orientação a objetos** foi criada para tentar aproximar o mundo real do mundo virtual: a idéia fundamental é tentar simular o mundo real dentro do computador.

- Para isso, nada mais natural do que utilizar Objetos, afinal, nosso mundo é composto de objetos, certo?!

Introdução Programação Orientada a Objetos

- Vamos a um exemplo prático: imagine que você está desenvolvendo um software para este colégio.
- O colégio tem diversos Alunos.
  - Como estamos tentando modelar um sistema baseado no sistema real, nada mais obvio do que existirem objetos do tipo Alunos dentro do nosso programa,
  - Estes serão os objetos que "simulam" as características e ações no mundo virtual que um cliente pode realizar no mundo real.

Introdução a OO

- POO é uma das principais inovações no desenvolvimento de software da última década.
- Complexidade: Principal problema na elaboração de programas.
  - Quão maior a complexidade, maior a possibilidade de haver erros.
- Erros de software têm elevado custo de correção
  - Podem causar situações de falhas operacionais;
  - Podem colocar vidas em perigo.

Propósito da POO

- POO foi desenvolvida devido às limitações existentes nas abordagens anteriores.
  - Projeto e programação estruturada.
- Para programas pequenos, não há necessidade de princípio organizacional.
- À medida que cresce o tamanho dos programas, torna-se difícil compreender programas que excedam a centenas de linhas de código.
  - Questão da complexidade

Introdução à POO – Programação Orientada a Objetos

Exercício 1:

- Imagine que você está desenvolvendo um software para gerenciar uma locadora de carros, assim identifique um conjunto de objetos que deverão ser modelados no sistema para representar o contexto do mundo real.