

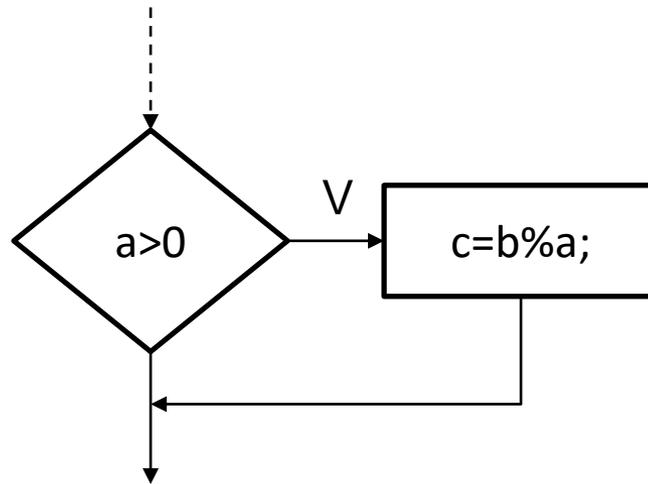


Estruturas de Repetição

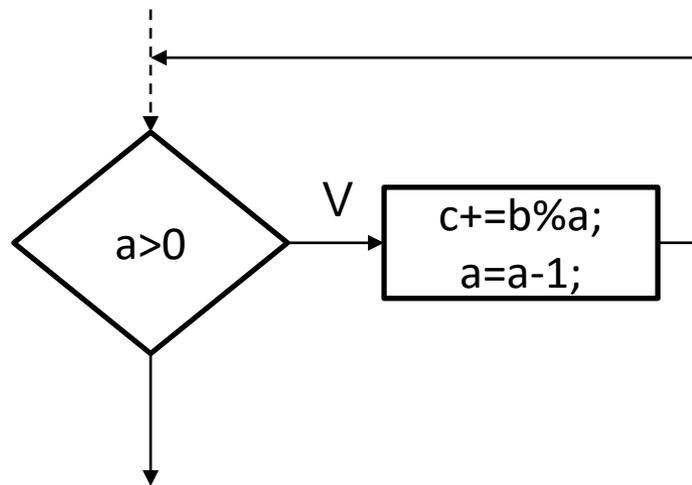
Prof. Rogério Neves

CMCC

Condicionais

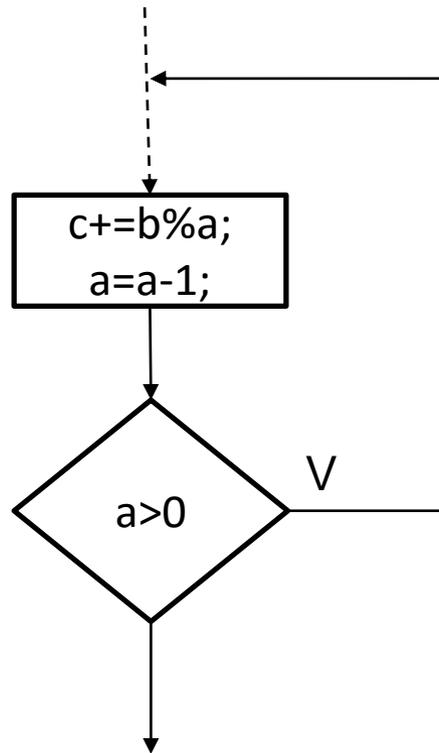


Laços de repetição



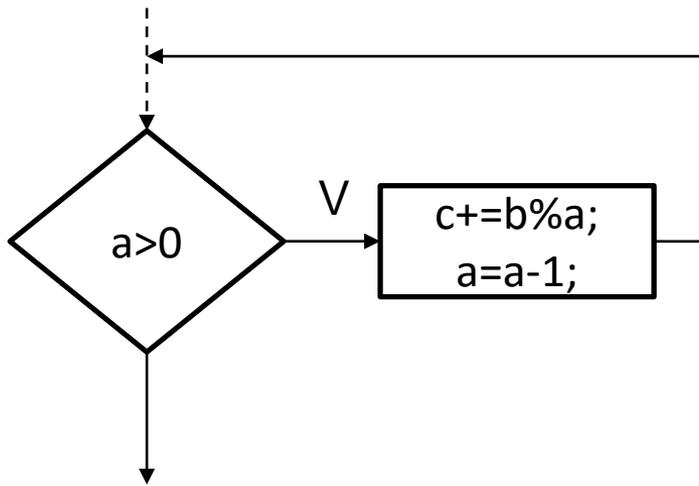
enquanto $(a > 0)$ faça { $c += b \% a;$ $a = a - 1;$ }

Laços de repetição

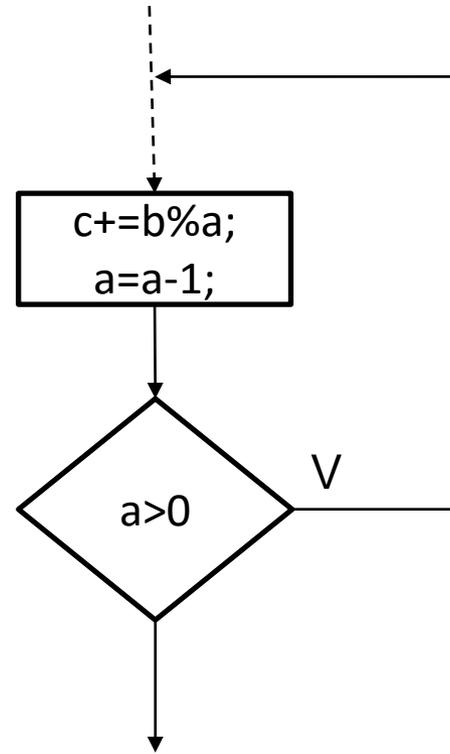


faça { `c += b%a;` `a = a - 1;` } enquanto (`a > 0`)

Laços em Java



```
while (condição) {  
    comandos ; ; ;  
}
```



```
do {  
    comandos ; ; ;  
} while (condição);
```

Para x de valor1 até valor2 faça

```
for (tipo var = valor_Inicial;  
     var < valor_Final;  
     operação de incremento/decremento)  
{  
    Comandos ; ; ;  
}
```

Exemplo 1:

```
for (int i = 10; i >0; i--) {  
    Comandos ; ; ;  
}
```

Para x de 1 até 10 faça

Exemplo 2:

```
for (double x = 1; x <=10; x=x+10) {  
    System.out.println(x);  
}
```

Exemplo 3:

```
for (float x = 1; x <= 10; x+=.01f) {  
    y=2*x*x-4*x-10;  
    System.out.printf("x=%6.2f\t y=%6.2f\n",  
                      x, y);  
}
```

Teste de Mesa

Qual o valor final de cada variável se executarmos o programa abaixo com os quatro diferentes tipos de <Laço-Exemplo> dados abaixo?

```
int contador = 5, a = contador, b = 0, c = 0;
```

<Laço-Exemplo n>

```
System.out.printf("a=%d\nb=%d\nc=%d\n\n", a, b, c);
```

Laço-Exemplo 1:

```
while (a > 0) {  
    c += a--;  
    b++;    }
```

Laço-Exemplo 2:

```
do {  
    c += a--;  
    b++;  
} while (a > 0);
```

Laço-Exemplo 3:

```
while (contador > 0) {  
    c += a--;  
    b++;  
}
```

Laço-Exemplo 4:

```
for (int i = 0; i < a; i++) {  
    c += i;  
    b++;  
}
```

Teste de Mesa

Qual o valor final de cada variável se executarmos o programa abaixo com os quatro diferentes tipos de <Laço-Exemplo> dados abaixo?

```
int contador = 5, a = 0, b = 0, c = 0;
```

<Laço-Exemplo n>

```
System.out.printf("a=%d\nb=%d\nc=%d\n\n", a, b, c);
```

Laço-Exemplo 1:

```
while (a > 0) {  
    c += a--;  
    b++;    }
```

Laço-Exemplo 2:

```
do {  
    c += a--;  
    b++;  
} while (a > 0);
```

Laço-Exemplo 3:

```
while (contador > 0) {  
    c += a--;  
    b++;  
}
```

Laço-Exemplo 4:

```
for (int i = 0; i < a; i++) {  
    c += i;  
    b++;  
}
```


Tabuada

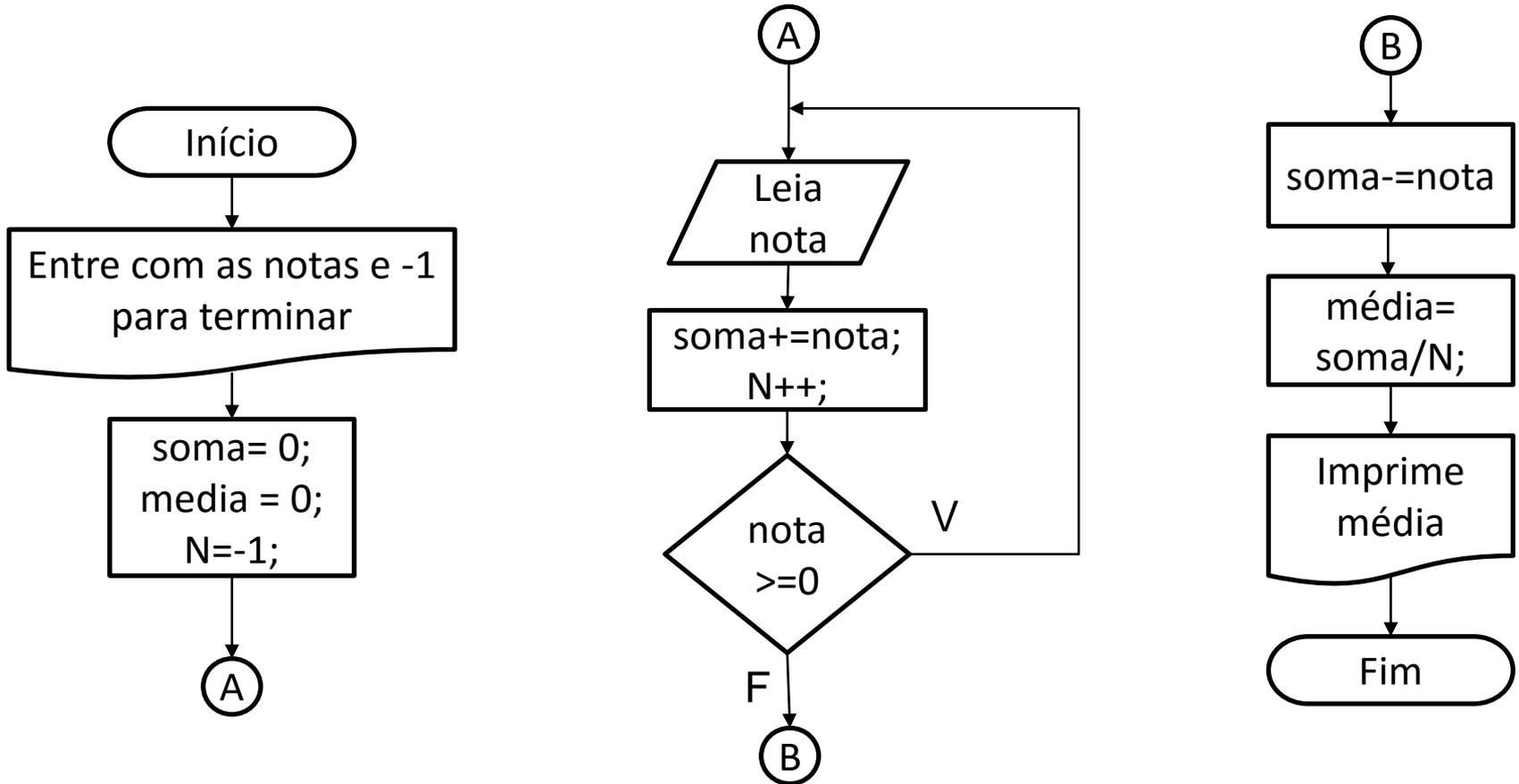
Laço:

```
int n = leia("Entre com a tabuada: ");
for (int i = 1; i <= 10; i++) {
    System.out.printf(
        "%3d x %3d = %4d\n", n, i, n * i);
}
```

Prática

1. Média de N alunos

Fazer utilizando console



Sequencial

Repetição

Sequencial

2. Juros compostos (no console)

- Dados (leia):
 - Valor inicial depositado
 - Complemento mensal
 - Juros mensais
 - Meses de aplicação
- Saída (Ex: R\$ 100, R\$ 50, 1% de juros, 3 meses)

Mês	depositado	juros no mês	juros total	Saldo
0	100	0.00	0.00	100.0
1	150	1.00	1.00	151.0
2	200	1.51	2.51	202.5
3	250	2.02	4.53	254.5

Saída formatada

Exemplo 2: R\$ 1000, R\$ 1000, 0,5% ao mês, 360 meses)

mês	depositado	juros no mês	juros total	saldo
0	1,000.00	0.00	0.00	1,000.00
1	2,000.00	5.00	5.00	2,005.00
2	3,000.00	10.02	15.02	3,015.02
3	4,000.00	15.08	30.10	4,030.10
4	5,000.00	20.15	50.25	5,050.25
5	6,000.00	25.25	75.50	6,075.50
6	7,000.00	30.38	105.88	7,105.88
7	8,000.00	35.53	141.41	8,141.41
8	9,000.00	40.71	182.12	9,182.12
9	10,000.00	45.91	228.03	10,228.03
10	11,000.00	51.14	279.17	11,279.17
11	12,000.00	56.40	335.56	12,335.56
12	13,000.00	61.68	397.24	13,397.24
...
...
359	360,000.00	4,992.61	644,515.04	1,004,515.04
360	361,000.00	5,022.58	649,537.62	1,010,537.62

Saída formatada

```
System.out.printf("Variaveis: %d, %.2f, %8s", byte1, x, myString);  
/*          "String      ↑      ↑      ↑ ", parâmetros (variáveis)  
                Recipientes
```

Recipientes (%) apropriados para cada formato:

%d - decimal inteiro	%u - inteiro positivo
%f - real formato xxx.yyyyyy	%E - real científico x.yyyyyyE+zzz
%c - caractere	%X - %d em hexadecimal maiúsculo
%s - String	%x - %d em hexadecimal minúsculo
%b - booleano	%o - %d em octal

Modificadores:

```
%[-][+][0-N][.0-9][lL][dxXuofeEgGcs]  
|| | | | | |  
|| | | | | +- formato (tabela acima)  
|| | | | +----- modificador long, ignorado  
|| | | +----- número de casas decimais após o .  
|| | +----- largura do campo (N caracteres)  
|| +----- mostra '+' para números positivos  
|+----- alinha a esq.  
+----- inicia formatação
```

Cada recipiente requer uma variável do tipo correspondente

*/

Precedencia de operadores	Em Java
Máxima precedência: separador, indexação, parâmetros, conversão de tipo	. [] () (tipo)
Multiplicativos: vezes, dividido, resto	* / %
Unários: positivo , negativo , negação (inversão bit a bit lógico), não (lógico), incremento e decremento pré-fixados	+n -n ~ ! ++expr --expr
Soma e subtração	+ -
de movimento (shift lógico)	<< >> >>>
Relacionais (lógicos): menor, maior, menor/maior ou igual	<> <= >=
de igualdade (lógico)	== !=
AND a nível de bit (bitwise AND)	&
OR exclusivo a nível de bit (bitwise exclusive OR)	^
OR inclusivo a nível de bit (bitwise inclusive OR)	
AND lógico	&&
OR lógico	
Ternarios Ex: X = (A>B?A:B)	(? :)
Incremento e decremento Pós-fixados	expr++ expr--
Atribuição	= += -= *= /= %= &= ^= = <<= >>= >>>=