



Programação 2

Instruções de controle

Prof. Domingo Santos
domingos.santos@upe.br

- Estruturas de controle:
 - Considerações Iniciais
 - Sequência
 - Seleção
 - Repetição

- Algoritmo:
 - Série de **ações** em uma **ordem específica**
 - Exemplo, ir trabalhar:
 - (1) Levantar da cama;
 - (2) tira o pijama;
 - (3) tomar banho;
 - (4) vestir-se;
 - (5) tomar café da manhã;
 - (6) dirigir o carro até o trabalho
 - Pseudocódigo:
 - Linguagem informal
 - Desenvolver algoritmos sem se preocupar com os estritos detalhes da sintaxe da linguagem
 - Exemplo:
 - Configure o total como zero
 - Configure o contador de notas como um
 - Enquanto contador de notas for menor ou igual a dez
 - Solicite para o usuário inserir a próxima nota
 - Insira a próxima nota
 - Adicione a nota ao total
 - Adicione um ao contador de notas
 - Configure a média da classe como o total dividido por dez
 - Exibe a média da classe
-

Estruturas de controle:

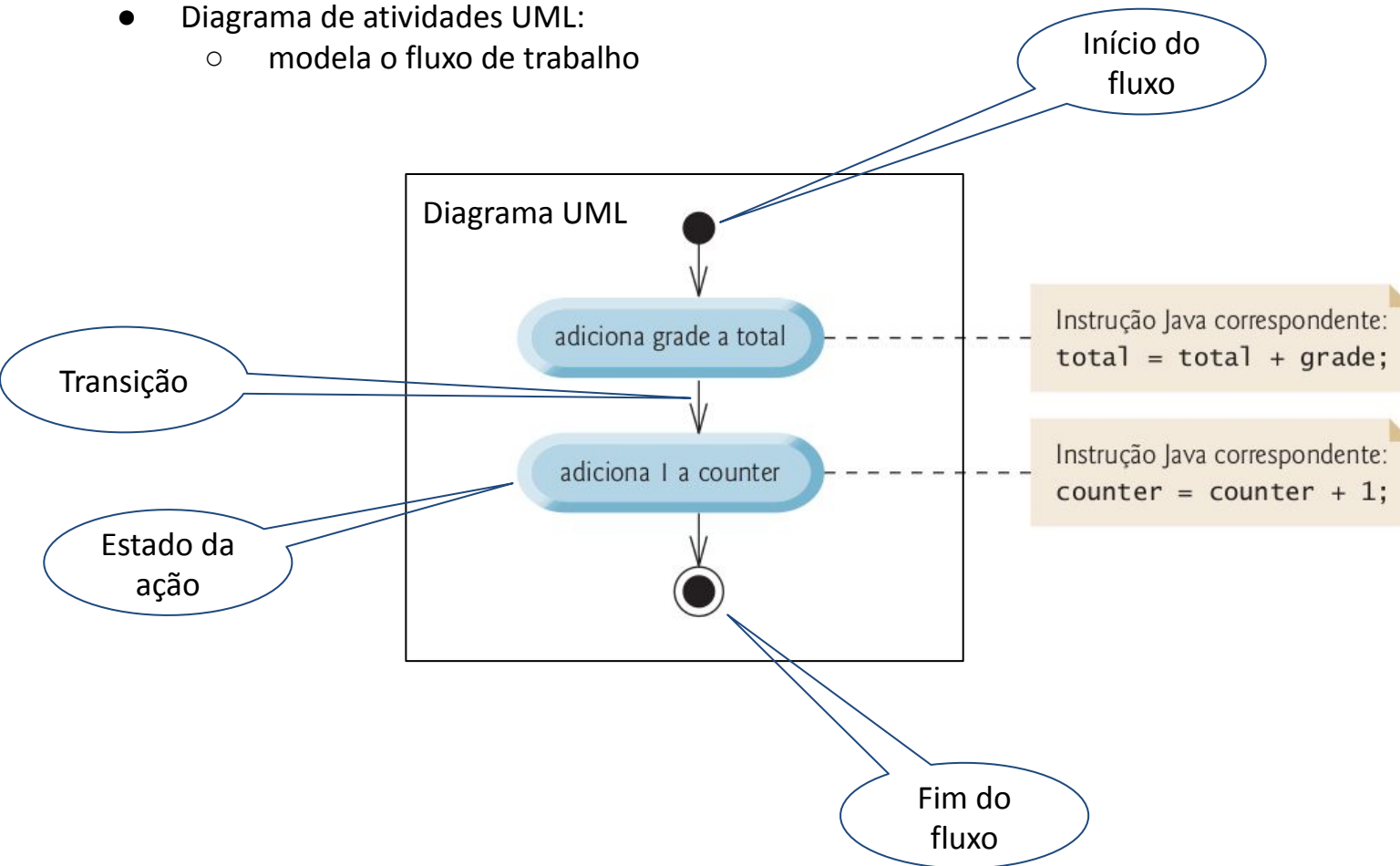
- Execução sequência: instruções em um programa são executadas uma após a outra na ordem em que são escritas
- Transferência de controle: especificação em que a próxima instrução a executar não é necessariamente a próxima
- Instrução goto:
 - No passado utilizado para realizar transferências de controle (feita de forma livre)
 - Raiz de muita dificuldade experimentada por grupos de desenvolvimento de software
 - o Java não contém uma instrução goto; entretanto, a palavra goto é reservada

Programação estruturada

- quase sinônimo de “eliminação de goto”
 - processo mais organizado, que viabiliza:
 - desenvolvimento mais rápido;
 - menos suscetíveis a bugs
 - maior facilidade de depuração
 - desenvolvimento focado em: **estrutura de sequência**, a **estrutura de seleção** e a **estrutura de repetição**
-

Estrutura de sequência em Java

- Leitura na ordem em que elas são escritas
- Diagrama de atividades UML:
 - modela o fluxo de trabalho



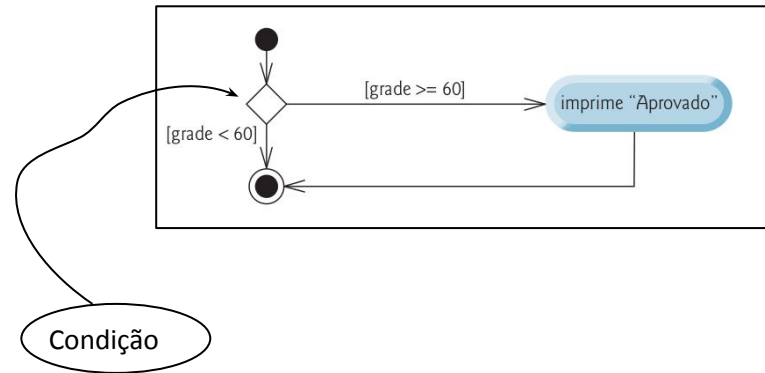
Criar cursos alternativos de ações (Única condição)

Pseudo código

Se a nota do aluno for maior que ou igual a 60

Imprime "Aprovado"

If em UML



If em Java

```
package com.example.helloworld;
public class conditions {
    public static void main(String[] args) {
        int studentGrade;
        studentGrade = 80;

        if (studentGrade >= 60)
            System.out.println("Passed");
    }
}
```

Sintaxe If

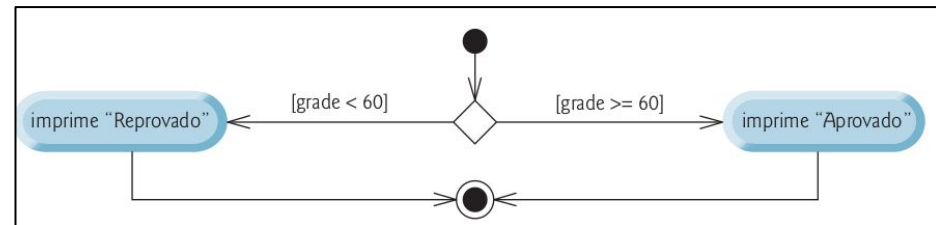
```
if (condição) {
    // bloco de código a ser executado
    se a condição for verdadeira
}
```

Criar cursos alternativos de ações (Múltiplas condições)

Pseudo código

Se a nota do aluno for maior que ou igual a 60
 Imprima "Aprovado"
Caso contrário
 Imprima "Reprovado"

If em UML



If em Java

```
package com.example.helloworld;
public class conditions {
    public static void main(String[] args) {
        int studentGrade;
        studentGrade = 80;

        if (studentGrade >= 60)
            System.out.println("Passed");
        else
            System.out.println("Failed");
    }
}
```

Sintaxe If else

```
if (condição) {
    // bloco de código a ser executado
    se a condição for verdadeira
}
else{
    // bloco de código a ser executado
    se a condição do if for falsa
}
```

Criar cursos alternativos de ações (Múltiplas condições, ifs dentro de ifs)

Pseudo código

```
Se a nota do aluno for maior que ou igual a 90
    Imprima "A"
caso contrário
    Se a nota do aluno for maior que ou igual a 80
        Imprima "B"
    caso contrário
        Se a nota do aluno for maior que ou igual a 70
            Imprima "C"
        caso contrário
            Se a nota do aluno for maior que ou igual a 60
                Imprima "D"
            caso contrário
                Imprima "F"
```

Sintaxe If else

```
if (condição) {
    // bloco de código a ser executado se a
    // condição for verdadeira
} else if (outra_condição) {
    // bloco de código a ser executado se a
    // primeira condição for falsa e a outra condição
    // for verdadeira
} else {
    // bloco de código a ser executado se nenhuma
    // das condições anteriores for verdadeira
}
```

If em Java

```
package com.example.helloworld;
public class conditions {
    public static void main(String[] args) {
        int studentGrade;
        studentGrade = 80;

        if (studentGrade >= 90)
            System.out.println("A");
        else if (studentGrade >= 80)
            System.out.println("B");
        else if (studentGrade >= 70)
            System.out.println("C");
        else if (studentGrade >= 60)
            System.out.println("D");
        else
            System.out.println("F");
    }
}
```


O problema do else oscilante

Exemplo de else oscilante (de qual if é o else?)

```
if (x > 5)
    if (y > 5)
        System.out.println("x and y are > 5");
else
    System.out.println("x is <= 5");
```

Utilização de chaves para deixar claro as hierarquias

```
if (x > 5)
{
    if (y > 5) {
        System.out.println("x and y are > 5");
    }
}
else {
    System.out.println("x is <= 5");
}
```

as chaves também permite a escrita de instruções em bloco

A estrutura de seleção múltipla switch

- realiza diferentes ações com base nos possíveis valores
- Pode ser utilizado com os tipos byte, short, int, char e String (Após Java SE 7)

Estrutura

```
switch (expressão) {  
    case valor1:  
  
        // bloco de código que será executado  
  
        break;  
  
    case valor2:  
  
        // bloco de código que será executado  
  
        break;  
  
    case valorN:  
  
        // bloco de código que será executado  
  
        break;  
  
    default:  
  
        // bloco de código que será executado se nenhum dos cases for aceito  
  
}
```

A estrutura de seleção múltipla switch

```
package com.example.helloworld;
public class ExemploSwitch {
    public static void main(String[] args) {
        int opcao = 2;
        switch (opcao) {
            case 1:
                System.out.println("Opção 1 selecionada");
                break;
            case 2:
                System.out.println("Opção 2 selecionada");
                break;
            case 3:
                System.out.println("Opção 3 selecionada");

            case 4:
                System.out.println("Opção 3 selecionada");
                break;

            default:
                System.out.println("Opção inválida");
        }
    }
}
```

O que será impresso se a variável opcao for:

- 2;
- -1
- 3
- 4

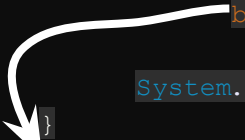
Instrução break

- Quando executada em um while, for, do...while ou switch, ocasiona a saída imediata desta instrução.
- A execução continua com a primeira instrução depois da instrução.

Exemplo com for de controle

```
package com.example.helloworld;
public class BreakExample {
    public static void main(String[] args)
    {
        // variável de controle também utilizada depois que loop termina
        int count;
        for (count = 1; count <= 10; count++) // faz o loop 10 vezes
        {
            if (count == 5)
                break; // termina o loop se a contagem for 5

            System.out.printf("%d ", count);
        }
        System.out.printf("\nBroke out of loop at count = %d\n", count);
    }
} // fim da classe BreakTest
```



Instrução continue


- Quando executada em um while, for, do...while ou switch
- Pula as instruções restantes no corpo do loop e prossegue com a próxima iteração do loop

Exemplo com for de controle

```
package com.example.helloworld;
public class BreakExample {
    public static void main(String[] args)
    {
        // variável de controle também utilizada depois que loop termina
        int count;
        for (count = 1; count <= 10; count++) // faz o loop 10 vezes
        {
            if (count == 5)
                continue; // pula o código restante no corpo do loop se
a contagem for 5

            System.out.printf("%d ", count);

        }
        System.out.printf("\nBroke out of loop at count = %d\n", count);
    }
} // fim da classe BreakTest
```



Operador E condicional (&&)

Possibilidades

expressão 1	expressão 2	expressão 1 && expressão 2
false	false	false
false	true	false
true	false	false
true	true	true

Exemplo “pode ou não dirigir”

```
public class ExemploOperadorAND {  
    public static void main(String[] args) {  
        int idade = 25;  
        boolean possuiCarteiraMotorista = true;  
  
        if (idade >= 18 && possuiCarteiraMotorista) {  
            System.out.println("Pode dirigir legalmente.");  
        } else {  
            System.out.println("Não pode dirigir legalmente.");  
        }  
    }  
}
```

Operador E condicional (&&)

Possibilidades

expressão 1	expressão 2	expressão 1 && expressão 2
false	false	false
false	true	false
true	false	false
true	true	true

Exemplo “verificar se o número é divisível por 2 e 3”

```
public class ExemploLoopComOperadorAND {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 10; i++) {  
            // qual seria a condição?  
        }  
    }  
}
```

Operador OU condicional (||)

Possibilidades

expressão 1	expressão 2	expressão 1 expressão 2
false	false	false
false	true	true
true	false	true
true	true	true

Exemplo “verificar se a nota do aluno é A”

```
public class ExemploLoopComOperadorOR {  
    public static void main(String[] args) {  
        int semesterAverage = 80;  
        int finalExam = 95;  
  
        if ((semesterAverage >= 90) || (finalExam >= 90))  
            System.out.println ("Student grade is A");  
  
    }  
}
```


Operador de negação lógica (!)

Possibilidades

expressão	!expressão
false	true
true	false

Exemplo “verificar se a nota do aluno é A”

```
public class ExemploOperadorAndNot {  
    public static void main(String[] args) {  
        int idade = 25;  
        boolean possuiCarteiraMotorista = true;  
  
        if (!(idade >= 18 && possuiCarteiraMotorista)) {  
            System.out.println("Não pode dirigir legalmente.");  
        } else {  
            System.out.println("Pode dirigir legalmente.");  
        }  
    }  
}
```

Operadores de atribuição compostos

Abreviam expressões de atribuição.

- $\text{variável} = \text{variável} \text{ operador expressão};$
- Termo $\text{operador} = +, -, *, / \text{ ou } \%$

expressões de atribuição	Abreviação
$\text{variável} = \text{variável} \text{ operador expressão}$	$\text{variável} \text{ operador} = \text{expressão};$
$c = c + 3;$	$c += 3$
$d = d - 4$	$d -= 4$
$e = e * 5$	$e *= 5$
$f = f / 3$	$f /= 3$

Operadores de incremento e decremento

- Operadores unários para adicionar 1 ou subtrair 1 do valor de uma variável numérica.

Operador	Nome do operador	Exemplo	Explicação
++	pré-incremento	++a	Incrementa a por 1, então utiliza o novo valor de a na expressão em que a reside.
++	pós-incremento	a++	Usa o valor atual de a na expressão em que a reside, então incrementa a por 1.
--	pré-decremento	--b	Decrementa b por 1, então utiliza o novo valor de b na expressão em que b reside.
--	pós-decremento	b--	Usa o valor atual de b na expressão em que b reside, então decrementa b por 1.

Exemplo

```
public class Increment
{
    public static void main(String[] args){
        // demonstra o operador de pós-incremento
        int c = 5;
        System.out.printf("c before postincrement: %d\n", c);
        System.out.printf("postincrementing c: %d\n", c++);
        System.out.printf(" c after postincrement: %d\n", c);
        System.out.println(); // pula uma linha
        // demonstra o operador de pré-incremento
        c = 5;
        System.out.printf(" c before preincrement: %d\n", c);
        System.out.printf("preincrementing c: %d\n", ++c);
        System.out.printf(" c after preincrement: %d\n", c);

    }
}
```

Instrução de repetição

While

Sintaxe

```
while (condição) {  
    // bloco de código a ser repetido enquanto a  
    condição for verdadeira  
}
```

Exemplo

```
int contador = 1;  
while (contador <= 5) {  
    System.out.println(contador);  
    contador++;  
}
```

Do While

```
do (condição) {  
    // bloco de código a ser repetido enquanto a  
    condição for verdadeira  
} while
```

Exemplo

```
int contador = 1;  
do {  
    System.out.println(contador);  
    contador++;  
} while (contador <= 5);
```

For

Sintaxe

```
for (inicialização; condição; atualização) {  
    // bloco de código a ser repetido enquanto a  
    condição for verdadeira  
}
```

Exemplo

```
for (int contador = 1; contador <= 5; contador++) {  
    System.out.println(contador);  
}
```

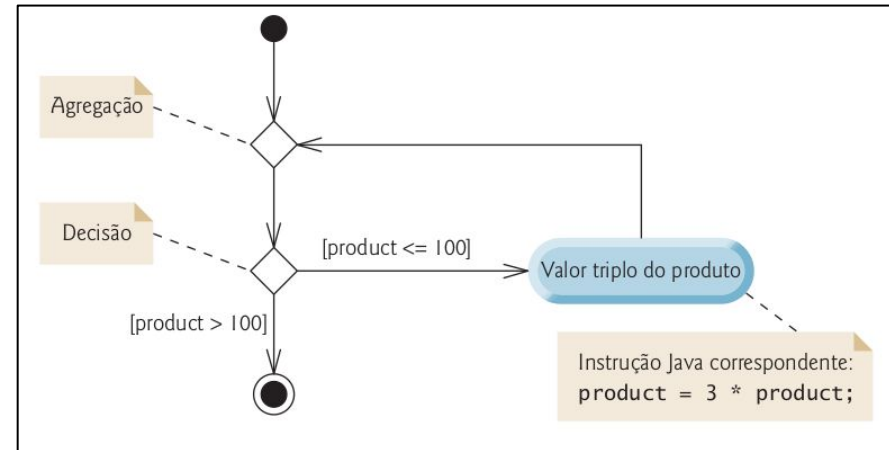
Instrução de repetição

While

Pseudo código

Enquanto houver mais itens em minha lista de compras
Comprar o próximo item e riscá-lo da minha lista

While em UML



While em Java

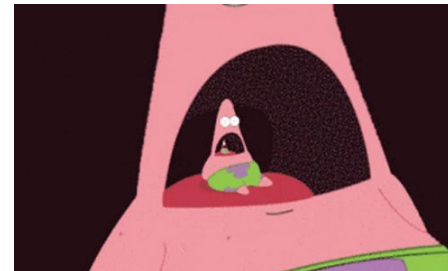
```
package com.example.helloworld;
public class WhileExamples {
    public static void main(String[] args) {
        int product;
        product = 1;

        while (product <= 100)
            product = 3 * product;

        System.out.printf("product: %d \n",
            product);
    }
}
```

Qual o problema se product for
inicializado com 0?

loop infinito



Repetição controlada por sentinela

Desenvolva um programa para tirar a média da classe que processe as notas de acordo com um número arbitrário de alunos toda vez que é executado. O processo deve terminar quando o valor de entrada for igual a 1

```
package com.example.helloworld;
import java.util.Scanner;
public class WhileExamples {
    public static void main(String[] args) {
        // cria Scanner para obter entrada a partir da janela de comando
        Scanner input = new Scanner(System.in);
        // fase de inicialização
        int total = 0; // inicializa a soma das notas inseridas pelo usuário
        int gradeCounter = 0; // inicializa nº da nota a ser inserido em seguida

        // fase de processamento
        // solicita entrada e lê a nota do usuário
        System.out.print("Enter grade or -1 to quit: " );
        int grade = input.nextInt();

        // fase de processamento utiliza repetição controlada por contador
        while (grade != -1) // faz o loop 10 vezes
        {
            total = total + grade; // adiciona grade a total
            gradeCounter = gradeCounter + 1; // incrementa counter
            // solicita entrada e lê a próxima nota fornecida pelo usuário
            System.out.print("Enter grade or -1 to quit: " );
            grade = input.nextInt();
        }

        // fase de término
        // se usuário inseriu pelo menos uma nota...
        if (gradeCounter != 0)
        {
            // usa número com ponto decimal para calcular média das notas
            double average = (double) total / gradeCounter; // operador de coerção

            // exibe o total e a média (com dois dígitos de precisão)
            System.out.printf("\nTotal of the %d grades entered is %d\n", gradeCounter,
total);

            System.out.printf("Class average is %.2f\n", average);
        }
        else // nenhuma nota foi inserida, assim gera a saída da mensagem apropriada
            System.out.println("No grades were entered");
    }
}
```

parte 1

parte 2

parte 3

Repetição controlada por contador

Considerando o seguinte problema: Uma classe de dez alunos se submeteu a um questionário. As notas (inteiros no intervalo 0–100) para esse questionário estão disponíveis. Determine a média da classe no questionário.

Aspectos

Variável de controle

Valor inicial

Incremento

Condição de
continuação do
loop

```
package com.example.helloworld;
import java.util.Scanner;
public class WhileExamples {
    public static void main(String[] args) {
        // cria Scanner para obter entrada a partir da janela de comando
        Scanner input = new Scanner(System.in);
        // fase de inicialização
        int total = 0; // inicializa a soma das notas inseridas pelo usuário
        int gradeCounter = 1; // inicializa nº da nota a ser inserido em seguida

        // fase de processamento utiliza repetição controlada por contador
        while (gradeCounter <= 10) // faz o loop 10 vezes
        {
            System.out.print("Enter grade: "); // prompt
            int grade = input.nextInt(); // insere a próxima nota
            total = total + grade; // adiciona grade a total
            gradeCounter = gradeCounter + 1; // incrementa o contador por 1
        }

        // fase de término
        int average = total / 10; // divisão de inteiros produz um resultado inteiro
        // exibe o total e a média das notas
        System.out.printf("\nTotal of all 10 grades is %d\n", total);
        System.out.printf("Class average is %d\n", average);
    }
}
```

Repetição controlada por contador

Considerando o seguinte problema: Uma classe de dez alunos se submeteu a um questionário. As notas (inteiros no intervalo 0–100) para esse questionário estão disponíveis. Determine a média da classe no questionário.

Aspectos

Variável de controle

Valor inicial

Incremento

Condição de
continuação do
loop

```
package com.example.helloworld;
import java.util.Scanner;
public class DoWhileExample {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        // fase de inicialização
        int total = 0; // inicializa a soma das notas inseridas pelo usuário
        int gradeCounter = 1; // inicializa n° da nota a ser inserido em seguida

        // fase de processamento utiliza repetição controlada por contador
        do // faz o loop 10 vezes
        {
            System.out.print("Enter grade: "); // prompt
            int grade = input.nextInt(); // insere a próxima nota
            total = total + grade; // adiciona grade a total
            gradeCounter = gradeCounter + 1; // incrementa o contador por 1
        } while (gradeCounter <= 10);

        // fase de término
        int average = total / 10; // divisão de inteiros produz um resultado inteiro
        // exibe o total e a média das notas
        System.out.printf("\nTotal of all 10 grades is %d\n", total);
        System.out.printf("Class average is %d\n", average);
    }
}
```


Diferenças entre do while e while

```
package com.example.helloworld;

public class WhileDiffs {
    public static void main(String[] args) {
        int i = 6;
        System.out.println("Usando do-while:");
        do {
            System.out.println(i);
            i++;
        } while (i <= 5);

        i = 6; // Reinicializa i para 1
        System.out.println("\nUsando while:");
        while (i <= 5) {
            System.out.println(i);
            i++;
        }
    }
}
```

Repetição controlada por contador

Considerando o seguinte problema: Uma classe de dez alunos se submeteu a um questionário. As notas (inteiros no intervalo 0–100) para esse questionário estão disponíveis. Determine a média da classe no questionário.

Aspectos

Variável de controle

Valor inicial

Incremento

Condição de
continuação do
loop

```
package com.example.helloworld;
import java.util.Scanner;
public class ForExample {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        // fase de inicialização
        int total = 0; // inicializa a soma das notas inseridas pelo usuário

        // fase de processamento utiliza repetição controlada por contador
        for (int gradeCounter = 1; gradeCounter <=10 ; gradeCounter++ ) // faz o loop 10 vezes
        {
            System.out.print("Enter grade: "); // prompt
            int grade = input.nextInt(); // insere a próxima nota
            total = total + grade; // adiciona grade a total
        }

        // fase de término
        int average = total / 10; // divisão de inteiros produz um resultado inteiro
        // exibe o total e a média das notas
        System.out.printf("\nTotal of all 10 grades is %d\n", total);
        System.out.printf("Class average is %d\n", average);
    }
}
```

Exemplos com a estrutura for

a) Varie a variável de controle de 1 a 100 em incrementos de 1.

```
for (int i = 1; i <= 100; i++)
```

b) Varie a variável de controle de 100 a 1 em decrementos de 1.

```
for (int i = 100; i >= 1; i--)
```

c) Varie a variável de controle de 7 a 77 em incrementos de 7.

```
for (int i = 7; i <= 77; i += 7)
```

d) Varie a variável de controle de 20 a 2 em decrementos de 2.

```
for (int i = 20; i >= 2; i -= 2)
```

e) Varie a variável de controle em relação aos valores 2, 5, 8, 11, 14, 17, 20.

```
for (int i = 2; i <= 20; i += 3)
```

f) Varie a variável de controle em relação aos valores 99, 88, 77, 66, 55, 44, 33, 22, 11, 0.

```
for (int i = 99; i >= 0; i -= 11)
```

Desafio para casa valendo meio ponto na nota final

- Atividade com 5 etapas
 - Enviar a atividade pelo classroom até dia 03/05
 - Atividade no classroom será chamada de Desafio 1
 - Respostas enviada por texto indicando a parte e a letra da resposta
-

Desafio para casa valendo meio ponto na nota final

parte 1 - Crie códigos (separados) que o usuário entre com um número(s) que exiba o:

- A. Fatorial de um Número
 - B. conversão de Graus Celsius para Fahrenheit
 - C. Calcular o Máximo entre Dois Números
-

Desafio para casa valendo meio ponto na nota final

parte 2- Suponha que $i = 1$, $j = 2$, $k = 3$ e $m = 2$. O que cada uma das seguintes instruções imprime?

- a) `System.out.println(i == 1);`
 - b) `System.out.println(j == 3);`
 - c) `System.out.println((i >= 1) && (j < 4));`
 - d) `System.out.println((m <= 99) & (k < m));`
 - e) `System.out.println((j >= i) || (k == m));`
 - f) `System.out.println((k + m < j) | (3 - j >= k));`
 - g) `System.out.println(!(k > m));`
-

parte 3 - O que o seguinte programa faz?

```
public class Printing
{
    public static void main(String[] args)
    {
        for (int i = 1; i <= 10; i++)
        {
            for (int j = 1; j <= 5; j++)
                System.out.print('@');
            System.out.println();
        }
    }
} // fim da classe Printing
```

Desafio para casa valendo meio ponto na nota final

parte 4 - O que o seguinte programa faz?

- Reproduzir cada figuras das letras abaixo
- Utilize
 - loops
 - `System.out.print('*');`
 - `System.out.print(' ');`

(a)

```
*
**
***
****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

(b)

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

(c)

```
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****
```

(d)

```
          *
         **
        ***
       ****
      *****
     ******
    *******
   *******
  *******
 *****
*****
```


Desafio para casa valendo meio ponto na nota final

parte 5 - Encontrar o Primeiro Número Divisível por 7 e 5 entre 1 e 100

Faça um programa utilizando break e continue, que percorra de 1 até 100, quando for divisível por 5 e 7 o programa deve terminar a execução