Programação de Aplicativos

ESTRUTURAS CONDICIONAIS E OPERADORES

Lorrany B Marim

SENAI

Pelo futuro do trabalho

Estruturas de Controle

- Execução sequencial: instruções são executadas na ordem em que aparecem por padrão.
- Transferência de controle: permite que a próxima instrução não seja a seguinte na sequência (desvia o fluxo).
- Problema histórico (anos 1960): uso indiscriminado de desvios (especialmente goto) gerou código confuso, difícil de manter e propenso a erros.
- Programação estruturada: movimento para eliminar goto; no Java, goto é palavra reservada (não usada).

Estruturas de Controle

- Teorema de Böhm–Jacopini: qualquer programa pode ser escrito sem goto, usando apenas três estruturas.

- Benefícios observados (anos 1970): menor tempo de desenvolvimento, mais prazos cumpridos, projetos dentro do orçamento, código mais claro, fácil de depurar e modificar, menos bugs.

- Três estruturas de controle:
  - Sequência: execução linear de instruções.
  - Seleção: escolha entre caminhos (em Java: if, if-else, switch).
  - Repetição: execução iterativa (em Java: while, do-while, for).

Operadores Relacionais

Operador	Nome	Exemplo	Resultado
= =	Igual	x = = 10	
!=	Diferente	3 != 2	true
<	Menor	10 < 10	false
>	Maior	10 > 6	true
>=	Maior ou igual	3 >= 3	true
<=	Menor ou igual	7 <= 6	false

Fonte: https://tsimpoo.wordpress.com/wp-content/uploads/2013/03/tabela-8-operadores-relacionais.png

Condicional Simples

- Usadas para escolher entre caminhos alternativos de execução conforme uma condição.
- Exemplo de critério:

  se nota >= 60,
  imprime “Aprovado”;
  caso contrário, segue o fluxo sem imprimir.

- Java:

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

Indentação: opcional no if, mas recomendada para evidenciar a estrutura do programa.

SENAI
Pelo futuro do trabalho

Condicional Composto

- if (seleção única): executa a ação apenas quando a condição é verdadeira; caso contrário, a ação é pulada.
- if...else (seleção dupla): define duas ações mutuamente exclusivas — uma para condição verdadeira e outra para condição falsa.
- Exemplo lógico (nota ≥ 60):

  imprime “Aprovado” se nota >= 60;
  caso contrário, imprime “Reprovado`.

- Fluxo após a decisão: independente do ramo, a execução continua na próxima instrução da sequência.
- O mapeamento é direto e reforça o uso didático do pseudocódigo.
- Indentação: o corpo do else também deve ser recuado; mantenha um padrão consistente de formatação em todo o programa.

SENAI
Pelo futuro do trabalho

Condicional Composto

- Java:

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

- O mapeamento é direto e reforça o uso didático do pseudocódigo.

- Indentação: o corpo do else também deve ser recuado; mantenha um padrão consistente de formatação em todo o programa.

Operadores Lógicos

- Estruturas que dependem de condição: if, if...else, while, do... while, for controlam o fluxo com base em uma condição.

- Condições simples: testam uma única proposição, usando operadores relacionais >, <, >=, <= e de igualdade ==, != (ex.: count <= 10, number != sentinelValue, total > 1000).

- Múltiplas condições: podem ser tratadas com instruções separadas ou if/if...else aninhados; quando o fluxo exige, recorre-se a condições mais complexas.

Operadores Lógicos AND (E)

- Objetivo do (E condicional): garantir que duas condições sejam ambas verdadeiras antes de executar um bloco.

- Símbolo: &&

- Java:

```
if (studentGrade >= 60 && studentGrade<=100){
    System.out.println("Passed");
}else{
    System.out.println("Failed");
}
```

Operadores Lógicos OR (OU)

- Operador (OU condicional): escolhe o caminho quando qualquer das condições for verdadeira (basta uma ser true).

- Símbolo: ||

- Java:

```
if (studentGrade >= 60 || studentGrade<=100){
    System.out.println("Passed");
}else{
    System.out.println("Failed");
}
```

Operadores Lógicos NOT (Não)

- Operador (NÃO lógico): inverte o valor lógico de uma condição (unário: atua sobre uma condição).
- Diferença para outros lógicos: &&, ||, &, |, ^ são binários (combinam duas condições); ! é unário.
- Símbolo: !
- Java:

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

SENAI
Pelo futuro do trabalho

Switch Case

- switch (seleção múltipla): escolhe a ação conforme o valor de uma expressão integral constante (byte, short, int, char).
- Expressão é avaliada uma vez; a execução entra no case correspondente.
- case usa constantes; default é opcional.
- Sem break, ocorre fall-through (continua nos próximos casos).
- break (em switch e laços): ao executar dentro de while, for, do...while ou switch, sai imediatamente dessa estrutura.

Switch Case - Exemplo

```
int diaDaSemana = 3;
    switch (diaDaSemana) {
        case 1:
            System.out.println("Domingo");
            break; // 'break' é importante para sair do switch após encontrar a correspondência
        case 2:
            System.out.println("Segunda-feira");
            break;
        case 3:
            System.out.println("Terça-feira");
            break;
        case 4:
            System.out.println("Quarta-feira");
            break;
        case 5:
            System.out.println("Quinta-feira");
            break;
        case 6:
            System.out.println("Sexta-feira");
            break;
        case 7:
            System.out.println("Sábado");
            break;
        default: // O 'default' é executado se nenhum dos 'cases' corresponder
            System.out.println("Número de dia inválido.");
    }
```

Switch Case - Moderno

```java
int diaDaSemana = 3;
    switch (diaDaSemana) {
        case 1 -> System.out.println("Domingo");
        // 'break' é importante para sair do switch após encontrar a correspondência
        case 2 -> System.out.println("Segunda-feira");
        case 3 -> System.out.println("Terça-feira");
        case 4 -> System.out.println("Quarta-feira");
        case 5 -> System.out.println("Quinta-feira");
        case 6 -> System.out.println("Sexta-feira");
        case 7 -> System.out.println("Sábado");
        default -> // O 'default' é executado se nenhum dos 'cases' corresponder
            System.out.println("Número de dia inválido.");
    }
```

SENAI
Pelo futuro do trabalho