

The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

ORACLE

Academy

Java Foundations

5-3

Instrução switch

ORACLE
Academy



Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

Objetivos

- Esta lição abrange os seguintes objetivos:
 - Criar uma estrutura de controle switch
 - Comparar construções if/else com estruturas de controle switch
 - Entender a finalidade da palavra-chave break



Como Usar uma Instrução if/else?

- Considere o cenário em que você precisa escrever um programa Java para implementar o seguinte:
 - O usuário insere uma nota entre 9 e 12, e o programa imprime o nome da nota
- Primeiro, vamos começar com uma solução usando uma instrução if/else

Solução: instrução if/else

```
Scanner in = new Scanner(System.in);
System.out.println("Enter your grade");
int grade = in.nextInt();
if (grade == 9){
    System.out.println("You are a freshman");
}
else if (grade == 10) {
    System.out.println("You are a sophomore");
}
else if (grade == 11) {
    System.out.println("You are a junior");
}
else if (grade == 12) {
    System.out.println("You are a senior");
}
else {
    System.out.println("Invalid grade");
} //endif
```

Condições complexas com uma construção if encadeada tendem a ser confusas e difíceis de serem mantidas

O exemplo de código a seguir mostra um if encadeado para determinar o nome da classe do aluno.

A Instrução switch

- A instrução switch fornece uma sintaxe mais eficiente para escolher entre várias alternativas

```
switch (<variable or expression>) {  
    case <literal value>: //code_block1  
                        [break;]  
    case <literal value>: // code_block2  
                        [break;]  
    default: //default_code  
} //end switch
```

A sintaxe da construção switch é mostrada no slide.

A palavra-chave switch indica uma instrução switch.

variável é a variável cujo valor você deseja testar. Se preferir, você pode usar uma expressão. A variável (ou o resultado da expressão) só pode ser do tipo char, byte, short, int ou String.

A palavra-chave case indica um valor que você está testando. Uma combinação da palavra-chave case com um valor literal é referida como um rótulo case.

A instrução break é uma palavra-chave opcional que faz com que a execução do código saia da instrução switch imediatamente.

Solução: instrução switch

```
Scanner in = new Scanner(System.in);
System.out.println("What grade are you in?");
int grade = in.nextInt();
switch (grade) {
    case 9:
        System.out.println("You are a freshman");
        break;
    case 10:
        System.out.println("You are a sophomore");
        break;
    case 11:
        System.out.println("You are a junior");
        break;
    case 12:
        System.out.println("You are a senior");
        break;
    default:
        System.out.println("Invalid grade");
} //end switch
```

Se comparada com a solução fornecida por uma instrução if , essa solução é mais compacta e legível.

A Instrução switch

- Se comparada com a instrução if/else, a instrução switch:
 - É mais simplificada que as instruções if encadeadas
 - É mais fácil de ser lida e mantida
 - Simplifica a organização de várias ramificações de código que podem ser executadas
 - Oferece um melhor desempenho
 - Pode ser usada para condições mais complexas

Quando Usar Construções switch

- Use quando você estiver testando:
 - Igualdade (não um intervalo)
 - Um valor único
 - Para valores conhecidos fixos no tempo de compilação
 - int, short, byte, char ou String

```
int month = 8;
month = in.nextInt();

switch (month) {
    case 1: case 3: case 5: case 7:
    case 8: case 10: case 12: System.out.print("31 days");
                                break;
    case 2: if(isLeapYear)){
        ..
    }
```

Só é possível testar um valor individual

Valores conhecidos

Se você não conseguir encontrar valores para casos de teste individuais, use uma construção if/else .

String em uma instrução switch: Exemplo

```
String typeOfDay;  
String dayOfWeekArg = "Thursday";  
  
switch (dayOfWeekArg) {  
    case "Monday": typeOfDay = "Start of work week";  
                    break;  
    case "Tuesday":  
    case "Wednesday":  
    case "Thursday": typeOfDay = "Midweek";  
                    break;  
    case "Friday": typeOfDay = "End of work week";  
                    break;  
    case "Saturday":  
    case "Sunday": typeOfDay = "Weekend";  
                    break;  
    default: System.out.print("Invalid");  
}//end switch
```

Este exemplo mostra como usar uma switch em uma expressão da instrução switch e em expressões de rótulo case.

Exercício 1

- Crie um novo projeto e adicione o arquivo `SwitchEx1.java` a ele
- Modifique `SwitchEx1.java` para implementar o seguinte com a instrução `switch`
 - O usuário insere o mês como um número
 - O nome do mês correspondente deve ser exibido
 - No caso de um mês inválido, a saída deve ser exibida como “Mês inválido”

Instrução switch: palavras-chaves

- As palavras-chaves a seguir podem ser usadas em uma instrução switch :
 - **switch**: especifica a variável para testar o valor.
 - **case**: compara o valor da variável switch.
 - **default**: quando a entrada não corresponde aos casos, a instrução default é executada. No entanto, ela é opcional.
 - **break**: é usada como a última instrução em cada lista de instruções case e faz com que o controle seja transferido para o fim da instrução switch

O que É uma Palavra-chave break?

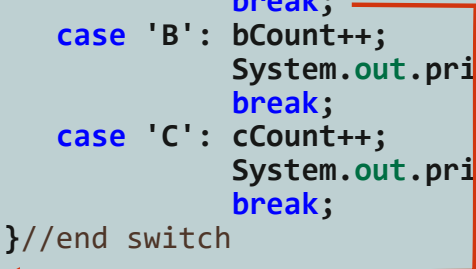
- É usada como a última instrução em cada lista de instruções case e faz com que o controle seja transferido para fora da instrução switch



O que É uma Palavra-chave break?

```
char option = 'A';
int aCount = 0, bCount = 0, cCount = 0;

switch (opção) {
    case 'A': aCount++;
              System.out.println("Count of A " + aCount);
              break;
    case 'B': bCount++;
              System.out.println("Count of B " + bCount);
              break;
    case 'C': cCount++;
              System.out.println("Count of C " + cCount);
              break;
} //end switch
//additional code . . .
```



Quando o valor da opção é A:

- O controle passa para a primeira instrução case.
- O valor de aCount++ é incrementado em 1.
- Devido à instrução break usada nesta instrução case, o controle é transferido para fora da instrução switch, e as outras duas instruções case não são executadas.

Saída: Contagem de A 1

Exercício 2

- Adicione o arquivo `SwitchEx2.java` ao projeto que você criou para o exercício 1
- Observe `SwitchEx2.java` e execute o programa
- Observe a saída

Exercício 2

- Modifique a instrução switch da seguinte forma:
- Remova as instruções break do caso 'A'.
 - Execute o programa
 - Observe a saída
- Remova as instruções break dos casos 'A' e 'B'
 - Execute o programa
 - Observe a saída

O que É switch Fall Through?

- switch fall through é uma condição que ocorrerá se não houver instruções break no fim de cada instrução case
- Todas as instruções depois do rótulo case correspondente são executadas em sequência, independentemente da expressão dos rótulos case subsequentes, até ser encontrada uma instrução break.

Entendendo o switch Fall Through

- Saída Esperada:

- Os valores das variáveis de contagem são incrementados em 1

```
char option = 'A';
int aCount = 0, bCount = 0, cCount = 0;

switch (opção) {
    case 'A': aCount++;
              System.out.println("Count of A " + aCount);

    case 'B': bCount++;
              System.out.println("Count of B " + bCount);
              break;

    case 'C': cCount++;
              System.out.println("Count of C " + cCount);
} //end switch
```

Nenhuma instrução break. Então, a execução continua com a próxima instrução case

Neste exemplo, se o valor da opção for A, ele corresponderá à primeira instrução case. Como não existe uma instrução break, a execução continua com a próxima instrução case até ser encontrada uma instrução break. O valor de bCount também é incrementado em 1.

Saída:

Count of A 1

Count of B 1

switch Fall Through: Exemplo

```
int month = 12;
switch (month) {
    case 2: System.out.println("28 days (29 in leap years)");
            break;
    case 4:
    case 6:
    case 9:
    case 11: System.out.println("30 days");
            break;
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 12: System.out.println("31 days");
            break;
    default: System.out.println("Illegal month number");
            break;
} //end switch
```

Este exemplo mostra como fall through é muito útil em alguns cenários. Às vezes, é preferível ter várias classes sem instruções break entre elas.

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Criar uma estrutura de controle switch
 - Comparar construções if/else com estruturas de controle switch
 - Entender a finalidade da palavra-chave break



ORACLE
Academy

JFo 5-3
Instrução switch

Copyright © 2022, Oracle e/ou suas empresas afiliadas. Oracle, Java e MySQL são marcas comerciais registradas da Oracle Corporation e/ou de suas empresas afiliadas. Outros nomes podem ser marcas comerciais de seus respectivos proprietários.

20

The Oracle Academy logo is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

ORACLE

Academy