

BCC201 – Introdução à Computação

Turmas 61, 62, 63, 64, 65 e 66



UFOP

Universidade Federal
de Ouro Preto

Puca Huachi Vaz Penna /
José Américo Trivellato
Messias

Departamento de Computação
Universidade Federal de Ouro Preto

<http://www.decom.ufop.br/puca>
puca@iceb.ufop.br

Aula 5

Aula Anterior



- Estruturas de Repetição: `if-else`



UFOP

Calculadora em C++

```
1 // Calculadora básica
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     char op;
8     double n1, n2, r;
9
10    cout << "Digite o primeiro número: ";
11    cin >> n1;
12    cout << "Digite o segundo número: ";
13    cin >> n2;
14    cout << "Digite a operação: ";
15    cin >> op;
```

Calculadora em C++ (continuação)

```
1
2     if (op == '+')
3         r = n1 + n2;
4     else
5         if (op == '-')
6             r = n1 - n2;
7         else
8             if (op == '*')
9                 r = n1 * n2;
10            else
11                r = n1 / n2;
12    cout << n1 << op << n2 << " = " << r << endl;
13    return 0;
14 }
```

Como melhorar o programa?



- No exemplo anterior se for digitado alguma operação diferente das 4 solicitadas, a divisão sempre será executado.
- Existem algumas possibilidades para melhorar o código. Em seguida, uma será apresentada, tente pensar em outras.



Calculadora em C++ (trecho melhorado)

```

1  if (op == '+')
2      r = n1 + n2;
3  else
4      if (op == '-')
5          r = n1 - n2;
6      else
7          if (op == '*')
8              r = n1 * n2;
9          else
10             if ( op == '/')
11                 r = n1 * n2;
12             else
13                 cout << "operação inválida!" << endl;
14  if (op == '+' || op == '-' || op == '*' || op ==
15      '/')
16      cout << n1 << op << n2 << " = " << r << endl;
17  return 0;
    }
    
```

Alternativa ao uso do `if`



- No exemplo anterior vimos que a variável `op` pode assumir diversos valores e que o uso dos `if` aninhados pode ser trabalhoso.
- Neste caso, temos um comando para verificar várias possibilidades para valores de uma variável ou expressão.
- Em C++, o comando `switch`.



Sintaxe do switch



```
1      switch (op) {  
2          case valor1:  
3              comandos1;  
4          break;  
5          case valor2:  
6              comandos2;  
7          ...  
8          default:  
9              comandosN;  
10     }
```



Funcionamento do switch



- O `switch` verifica se a expressão é equivalente ao `valor1`,
- se for, executa o grupo de `comandos1` até que encontre um `break`.
- Se a expressão não for igual a `valor1`, verifica a relação com `valor2` e assim por diante.
- Se não for nenhum dos valores, o grupo de `comandosN` do `default` é executado.



Calculadora em C++ (equivalente ao uso do if)



```
1      switch (op) {
2          case '+':
3              r = n1 + n2;
4              break;
5          case '-':
6              r = n1 - n2;
7              break;
8          case '*':
9              r = n1 * n2;
10             break;
11          case '/':
12              r = n1 / n2;
13             break;
14          default:
15              cout << "operação inválida!" << endl;
16              invalido = true;
17      }
```

Observações sobre o `switch`



- O `switch` só permite comparar expressões com constantes.
- Se precisarmos comparar com variáveis ou verificar faixas de valores, devemos usar o comando `if`.
- Se não usamos o comando `break` em cada `case` o programa continuaria até o fim do bloco



```
1  switch (x) {  
2      case 1:  
3          // se x for 1, passa por aqui  
4      case 2:  
5          // se x for 1 ou 2, passa por aqui  
6      case 3:  
7          cout << "x é 1, 2 ou 3";  
8      break ;  
9      default :  
10         cout << "x não é 1, 2 ou 3";  
11 }
```



Exemplo 1



Escreva um programa que leia o código de um determinado produto e mostre a sua classificação. Utilize a tabela abaixo como referência:

Código	Descrição
1	Alimento não-perecível
2	Alimento perecível
3	Vestuário
4	Limpeza



```

1 //Imprime a classificação de um produto
2 int main() {
3     int cod;
4     cout << "Digite o código: "; cin >> cod;
5     switch (cod) {
6         case 1:
7             cout << "Alimento não-perecível";
8             break;
9         case 2:
10            cout << "Alimento perecível";
11            break;
12         case 3:
13            cout << "Vestuário";
14            break;
15         case 4:
16            cout << "Limpeza";
17            break;
18     }
19     return 0;
20 }
```

Exemplo 2



Dada uma letra, escreva na tela se essa letra é uma vogal ou consoante (considerar apenas letras minúsculas).



UFOP

```

1 //Verifica se uma letra é vogal ou consoante
2 int main() {
3     char letra;
4     cout << "Digite uma letra: ";
5     cin >> letra;
6
7     switch (letra)
8     {
9         case 'a':
10        case 'e':
11        case 'i':
12        case 'o':
13        case 'u':
14            cout << "Vogal" << endl;
15            break;
16        default:
17            cout << "Consoante" << endl;
18    }
19    return 0;
20 }

```


Exemplo 3



Escreva um programa que indique o número de dias existentes em um mês.

Número de dias	Mês
28	02
30	04, 06, 09, 11
31	01, 03, 05, 07, 08, 10, 12



```

1 //Imprime o número de dias do mês
2 int main() {
3     int mes;
4     cout << "Digite o número do mês: ";
5     cin >> mes;
6     switch (mes)
7     {
8         case 1: case 3: case 5: case 7:
9         case 8: case 10: case 12:
10            cout << "0 mês tem 31 dias" << endl;
11            break;
12        case 4: case 6: case 9: case 10:
13            cout << "0 mês tem 30 dias" << endl;
14            break;
15        case 2:
16            cout << "0 mês tem 28 dias" << endl;
17            break;
18    }
19    return 0;
20 }

```

Exercício



Faça um programa que leia as notas da primeira prova dos **60 alunos** de BCC201 e calcule e imprima a média das notas. O programa deve verificar se nota digitada está correta, ou seja, $0 \leq nota \leq 10$.



Próxima Aula



- Comando de repetição

