

# Estruturas seletivas

---

# Switch-case

---

- O comando `switch` é um comando de seleção múltipla;
- O `switch` testa sucessivamente o valor de uma expressão contra uma lista de constantes inteiras ou de caracteres:
  - Só funciona para INTEIROS e CARACTERES!
- O comando `switch` somente testa a igualdade da variável em relação a valores constantes, diferente do comando `if` que é capaz de testar expressões lógicas e/ou relacionais.

# Exemplo 1:

---

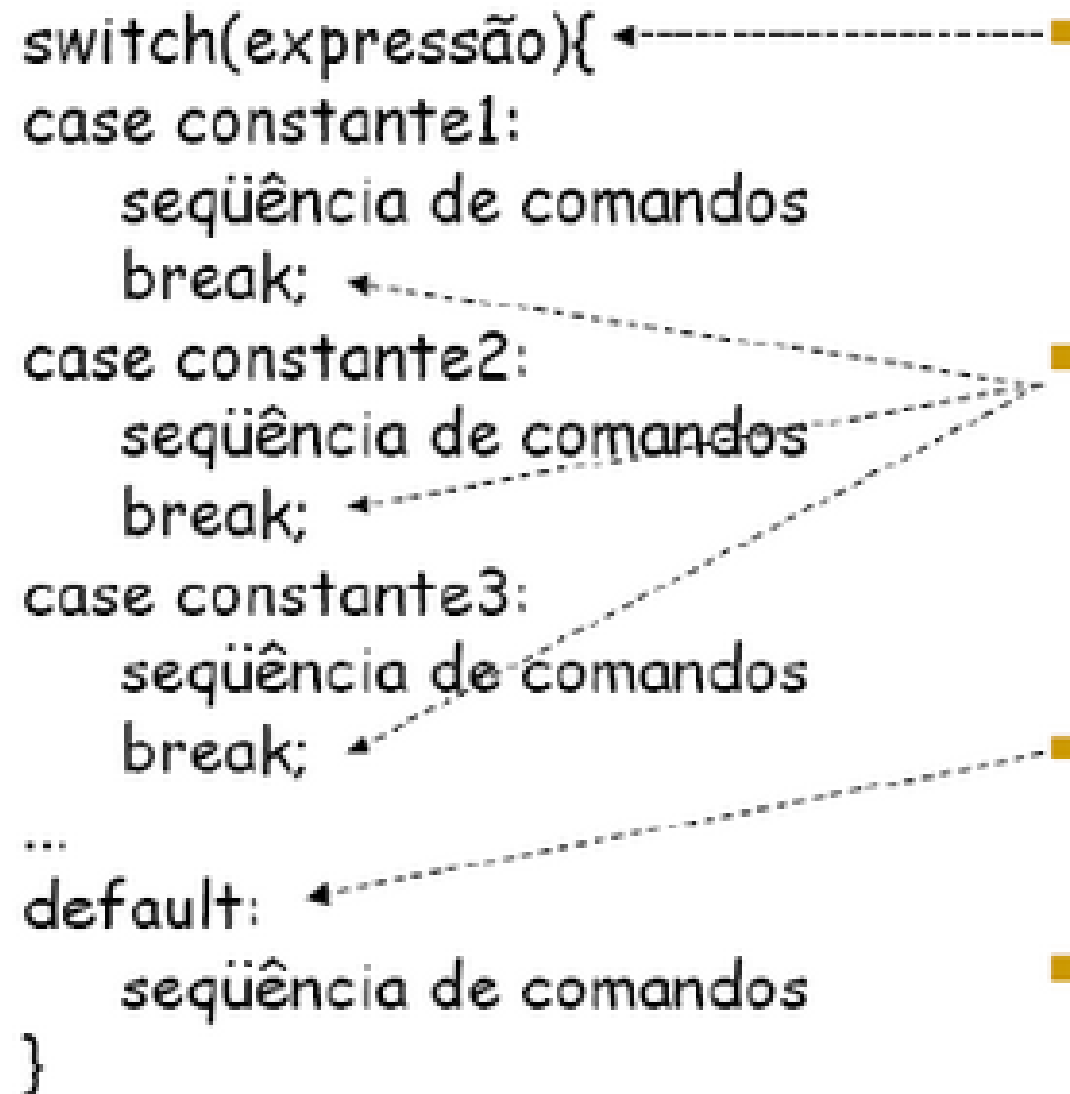
```
static void Main(string[] args)
{
    int num;
    Console.WriteLine("Digite um numero: ");
    num = int.Parse(Console.ReadLine());
    switch (num)
    {
        case 9:
            Console.WriteLine("o numero digitado é 9");
            break;
        case 10:
            Console.WriteLine("o numero digitado é 10");
            break;
        default:
            Console.WriteLine("o numero digitado não é 9 nem 10");
            break;
    }
}
```

- Obs. 1: abrimos e fechamos “{ }” ao iniciar um comando switch!
- Obs. 2: nos case, utilizamos “:” para abrir e break para terminar!!

# Switch-case

---

```
switch(expressão){  
  case constante1:  
    seqüência de comandos  
    break;  
  case constante2:  
    seqüência de comandos  
    break;  
  case constante3:  
    seqüência de comandos  
    break;  
  ...  
  default:  
    seqüência de comandos  
}
```



■ O valor da expressão é testado, na ordem, contra os valores das constantes especificadas nos comandos case;

■ Quando uma coincidência for encontrada, a seqüência de comando associada àquele case será executada até que o comando break ou o fim do comando switch seja alcançado;

■ O comando default é executado se se nenhuma coincidência for detectada;

■ O default é opcional.

# Exemplo 2

---

- Nos códigos...

## Exemplo 3

---

- Implemente na calculadora desenvolvida as seguintes funções:
  - Se a operação digitada for “+”, será efetuada a soma;
  - Se a operação digitada for “-”, será efetuada a subtração;
  - Se a operação digitada for “x”, será efetuada a multiplicação
  - Se a operação digitada for “/”, será efetuada a divisão;

# Operador ternário → ?

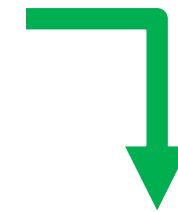
---

- Pode ser considerado uma variação simplificada de uma estrutura condicional `if-else`.
- Nesta estrutura, utilizamos o operador ternário “?”. Sua forma geral é:
  - `<condição> ? <instrução1> : <instrução2>`
- No caso acima, caso `condição` seja verdadeira, a `instrução1` é executada, caso seja falso, a `instrução2` é executada.

## Exemplo 4

---

```
static void Main(string[] args)
{
    int x = 0, i;
    i = int.Parse(Console.ReadLine());
    x = (i % 2 == 0) ? x = 1 : x = 2;
    Console.WriteLine("Resultado = "+x);
}
```



Traduzindo:

Se  $i \% 2 == 0$  for  
verdadeiro,  $x = 1$ , se não,  $x$   
 $= 2$



## Exemplo 4

---

```
static void Main(string[] args)
{
    int x = 0, i;
    i = int.Parse(Console.ReadLine());
    if (i % 2 == 0)
    {
        x = 1;
    }
    else
    {
        x = 2;
    }
    Console.WriteLine("Resultado = "+x);
}
```

# Exemplo 5

```
static void Main(string[] args)
{
    int a = 0, i;
    i = int.Parse(Console.ReadLine());
    a = i % 2 == 0 ? a=1 : a=-1;
    Console.WriteLine(i+" é "+((a < 0) ? "impar" : "par"));
}
```

Traduzindo:  
Se  $i \% 2 == 0$   
for verdadeiro,  
a=1, se não, a=-1

- Será mostrado um valor inteiro, no caso, a variável i;
  - Também será mostrado um texto , no caso:
    - Se  $a < 0$ , mostra "impar"
    - Se  $a > 0$ , mostra "par"

# Exemplo 6

---

```
static void Main(string[] args)
{
    int i;
    i = int.Parse(Console.ReadLine());
    Console.WriteLine(i+" é "+(i % 2 == 0 ? "par" : "impar"));
}
```