

Estruturas de repetição – parte 2

while

- O comando **while** permite implementar laços de repetição, ou seja, definir trechos do código de um programa que se repetem.
- É conhecido como comando `enquanto`.
- Neste comando, um bloco de instruções é executado repetidamente enquanto uma condição for verdadeira.
- A condição é testada antes da primeira execução do bloco de instruções.

while

- É similar a um laço `for` sem definição do valor inicial da variável contadora e sem incremento/decremento
- É necessário definir o valor inicial da variável contadora, antes do comando `while` iniciar.
- o incremento ou decremento da variável contadora deve ser feito dentro das instruções do laço de repetição `while`.

while

- Estrutura:

```
while(condição){  
  
    bloco de instruções  
  
}
```

- Facilita a implementação de laços com número de execuções indefinido.

while

- Enquanto a condição for verdadeira, o bloco de instruções é executado.
- Quando a condição for falsa, o laço de repetição não será executado, ou seja, a execução é desviada para o final.
- Se a condição já for falsa no momento do primeiro teste, o bloco nem chega a ser executado e o algoritmo continua a execução na instrução seguinte ao fim do `while`.

while vs for

- Um `for` dessa maneira:

```
for(i=0; i<10; i++){  
    //bloco de instruções  
}
```

- Equivale a um `while` assim:

```
i=0;  
while(i<10){  
    //bloco de instruções  
    i++;  
}
```

Exemplo 1

- Desenvolva um algoritmo que mostre os números de 0 a 100, utilizando o comando **while**.

Exemplo 1

```
static void Main(string[] args)
{
    int i = 0;
    while (i < 100)
    {
        Console.WriteLine("Contador = "+i);
        i++;
    }
}
```


Exemplo 2

- Desenvolva um algoritmo que mostre os números pares de 0 a 100, utilizando o comando **while**.

Exemplo 2

```
static void Main(string[] args)
{
    int par = 0;
    Console.WriteLine("Os números pares de 1 a 100: ");
    while (par <= 100)
    {
        Console.WriteLine("Contador = "+par);
        par+=2; //mesma coisa que par = par +2;
    }
}
```

Exemplo 3

- Desenvolva um algoritmo que leia continuamente o valor de um número inteiro chamado **op**, e então mostre na tela se **op** é positivo ou negativo. O programa deverá terminar quando o usuário digitar **zero** para **op**.

Exemplo 3

```
static void Main(string[] args)
{
    int op = 1;
    while (op != 0) //enquanto op for diferente de 0
    {
        Console.WriteLine("Digite um numero (0 para finalizar): ");
        op = int.Parse(Console.ReadLine());
        if (op > 0)
        {
            Console.WriteLine("Este numero é positivo\n");
        }
        else if (op < 0)
        {
            Console.WriteLine("Este numero é negativo\n");
        }
    }
}
```

Exemplo 4

- Desenvolva um algoritmo que leia continuamente um caractere pelo teclado chamado **Ch**, e só termine de fazer a leitura quando o usuário digitar a letra '**q**'.

Exemplo 4

```
static void Main(string[] args)
{
    char ch;
    ch = ' ';
    while (ch != 'q')
    {
        ch = Console.ReadKey(false).KeyChar;
        //Console.WriteLine("Tecla digitada = " + ch);
    }
}
```

Exemplo 4

```
static void Main(string[] args)
{
    char ch;
    ch = ' ';
    while (ch != 'q')
    {
        ch = Console.ReadKey(true).KeyChar;
        Console.WriteLine("Tecla digitada = " + ch);
    }
}
```

do while

- O comando **do while** permite novamente implementar laços de repetição, ou seja, definir trechos do código de um programa que se repetem.
- É conhecido como comando **repita enquanto** ou **faça enquanto**
- Estrutura:

```
do{  
    bloco de instruções;  
} while(condição);
```


do while

- Neste comando, um bloco de instruções é executado SEMPRE pelo menos 1 vez, até que a condição deixe de ser verdadeira.
- Isto acontece porque a condição é testada no final do laço de repetição (após a execução do bloco de instruções).
- Enquanto a condição for verdadeira, a execução volta para a execução no início do bloco (após o `do`) e o bloco de instruções volta a ser executado.
- Quando a condição for falsa, a execução do laço de iterações para de acontecer e a execução do algoritmo na próxima instrução após o teste da condição:
`while(condição);`

Exemplo 5

- Desenvolva um algoritmo que mostre os os números de 0 a 100, utilizando o comando **do... while**.

Exemplo 5

```
static void Main(string[] args)
{
    int i = 0;
    do //faça
    {
        Console.WriteLine(i);
        i++;
    } while (i<100); //enquanto i for menor que 100
}
```

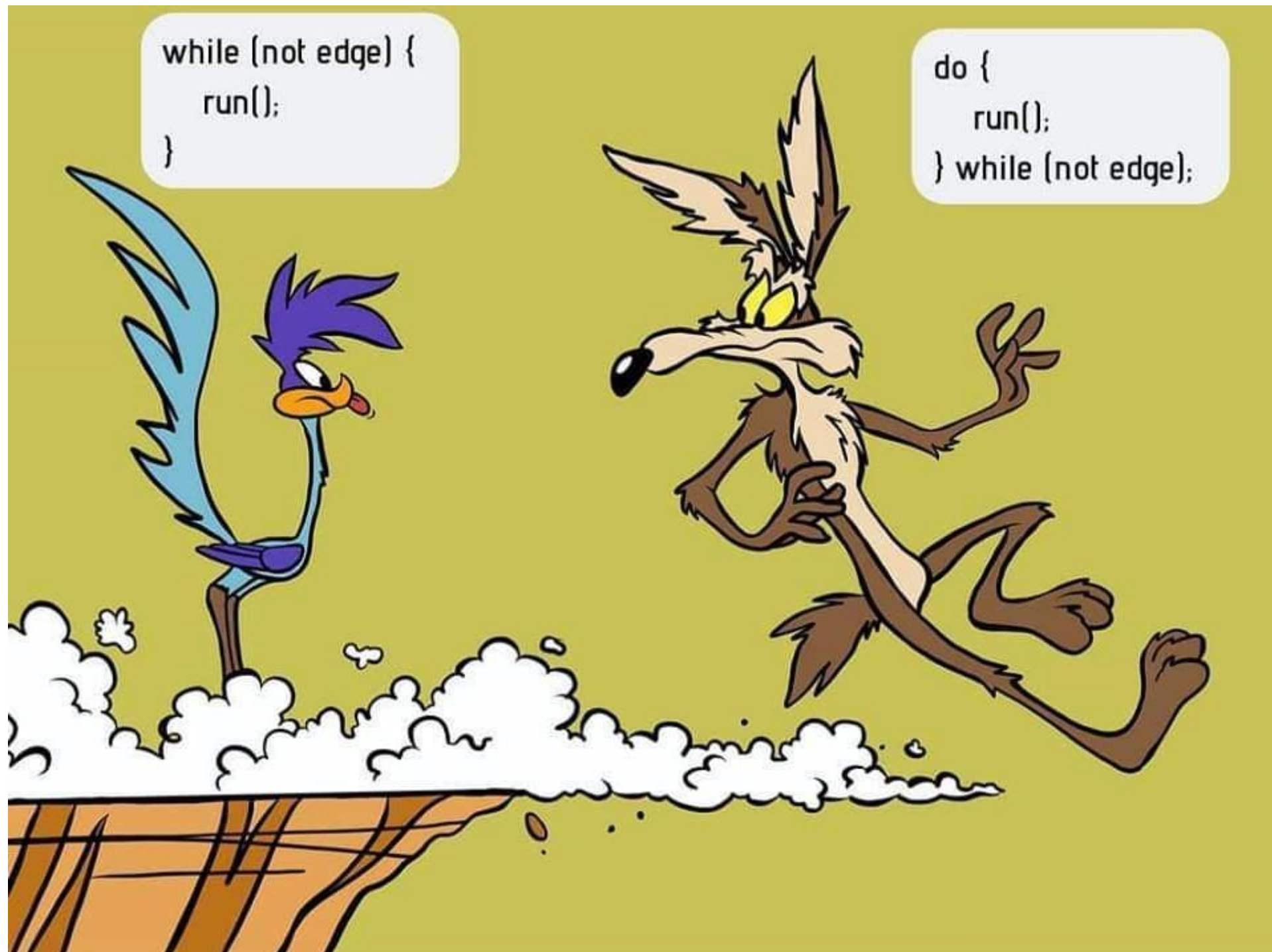
Exemplo 6

- Desenvolva um algoritmo que leia continuamente o valor de um número inteiro chamado **nr**, e então mostre na tela se **nr** é positivo ou negativo. O programa deverá terminar quando o usuário digitar **zero** para **nr**.

Exemplo 6

```
static void Main(string[] args)
{
    int nr;
    do //faça
    {
        Console.WriteLine("Digite um numero (0 para finalizar): ");
        nr = int.Parse(Console.ReadLine());
        if (nr % 2 == 0)
            Console.WriteLine("Este numero é par");
        else
            Console.WriteLine("Este numero é impar");
    } while (nr != 0); //enquanto nr for diferente de 0
}
```

do while



Exemplo 7

- Desenvolva um algoritmo que leia e some continuamente o valor de uma parcela, e pare de ler somente quando o valor digitado para a parcela for zero. Ao final, mostre a soma total das parcelas.

Exemplo 7

```
static void Main(string[] args)
{
    int soma = 0, parcela;
    do
    {
        Console.WriteLine("valor da parcela: (0 para finalizar): ");
        parcela = int.Parse(Console.ReadLine());
        soma += parcela;
    } while (parcela != 0);

    Console.WriteLine("Soma: "+soma);
}
```


Exemplo 8

- Desenvolva um algoritmo que peça para o usuário escolher entre as seguintes opções:
 1. mamão
 2. abacaxi
 3. laranja
- E em seguida mostre na tela a opção escolhida. Porém, caso o usuário digite uma opção inválida, a leitura deve ser feita novamente até que uma opção válida seja lida.

Exemplo 8

```
static void Main(string[] args)
{
    int i;
    do
    {
        Console.WriteLine("\nEscolha a fruta pelo numero:\n");
        Console.WriteLine("\t(1)...Mamão");
        Console.WriteLine("\t(2)...Abacaxi");
        Console.WriteLine("\t(3)...Laranja\n");
        i = int.Parse(Console.ReadLine());
    } while ((i < 1) || (i > 3));
    switch (i)
    {
        case 1:
            Console.WriteLine("\t\tVoce escolheu Mamão.\n");
            break;
        case 2:
            Console.WriteLine("\t\tVoce escolheu Abacaxi.\n");
            break;
        case 3:
            Console.WriteLine("\t\tVoce escolheu Laranja.\n");
            break;
    }
}
```