

Apresentação

Curso:

Programação .NET I

Aula 03:

Objetivos:

Estruturas de Repetição;

for;

while;

do-while;

for-each;

Estruturas de repetição

As estruturas de repetição, como o próprio pressupõe, são utilizadas para repetir um determinado bloco de código, respeitando uma expressão lógica a ser testada.

No C#, a princípio, temos três estruturas básicas de repetição: *while*, *for* e *do-while*.

Estruturas de repetição (while)

Escopo da estrutura while:

```
while(EXPRESSÃO LÓGICA)
{
    BLOCO DE INSTRUÇÕES
    *ATUALIZAÇÃO DO CONTROLE
}
```

ou:

```
while(EXPRESSÃO LÓGICA)
    LINHA DE INSTRUÇÃO;
```

Estruturas de repetição (while)

Apontamentos para a instrução while:

- A expressão precisa ser uma expressão booleana;
- A expressão booleana deve ser escrita entre parênteses;
- Se a instrução booleana for avaliada como falsa na primeira avaliação, o bloco de instruções não será executado;
- Se quiser executar somente UMA instrução no while, não há necessidade de chaves.

Estruturas de repetição (while)

O que vai acontecer se o seguinte código for escrito?

```
...  
while(true)  
{  
    ...  
    //BLOCO DE INSTRUÇÕES  
    ...  
}  
...
```

Estruturas de repetição (while)

O que vai acontecer se o seguinte código for escrito?

```
...  
while(true)  
{  
    //BLOCO DE INSTRUÇÕES  
    break;  
}  
...
```

Exercício Prático (while)

Exercício 1a:

Name: estrutura_while

Solution name: Nome_Sobrenome-Aula03

Façam um código para ler um inteiro e retornar o nome do dia da semana equivalente (Aula02:estrutura_if/switch).

Altere-o para repetir o preenchimento, enquanto o usuário colocar um dado válido (dentro do intervalo 1-7), caso contrário, escreva “Desconhecido” e encerre o programa (deixem o “PAUSE” antes de encerrar).

1 = Domingo ... 7 = Sábado

Resposta - Exercício Prático (while)

```
string nomeDiaSemana = "";
bool flag = true;

while (flag)
{
    Console.WriteLine("Programa de estrutura while");
    Console.WriteLine("Digite um numero inteiro entre 1 e 7");
    int diaSemana = Int32.Parse(Console.ReadLine());

    if (diaSemana == 1)
        nomeDiaSemana = "Domingo";
    else if (diaSemana == 2)
        nomeDiaSemana = "Segunda-feira";
    else if (diaSemana == 3)
        nomeDiaSemana = "Terça-feira";
    else if (diaSemana == 4)
        nomeDiaSemana = "Quarta-feira";
    else if (diaSemana == 5)
        nomeDiaSemana = "Quinta-feira";
    else if (diaSemana == 6)
        nomeDiaSemana = "Sexta-feira";
    else if (diaSemana == 7)
        nomeDiaSemana = "Sábado";
    else
    {
        nomeDiaSemana = "Desconhecido...\n\rEncerrando o programa!";
        flag = false;
    }

    Console.WriteLine("Dia escolhido: " + nomeDiaSemana + "\n\rAperte uma tecla para continuar...");
    Console.ReadKey(true);
    Console.Clear();
}
```



Exercício Prático (while)

Exercício 1b:

Name: estrutura_while_dez

Solution: Add to solution

Copie o código do exercício anterior e altere a repetição para ser feita por 10 vezes. Mostre em que passo a leitura se encontra.

1 = Domingo ... 7 = Sábado

- O programa só finaliza depois de ler 10 vezes.
- As mensagens (nomes dos dias) devem ser exibidas após cada leitura (normalmente, como era antes!).

Resposta - Exercício Prático (while)

```
string nomeDiaSemana = "";
int leitura = 1;

while (leitura <= 10)
{
    Console.WriteLine("Programa de estrutura while - Leitura: " + leitura);
    Console.WriteLine("Digite um numero inteiro entre 1 e 7");
    int diaSemana = Int32.Parse(Console.ReadLine());

    if (diaSemana == 1)
        nomeDiaSemana = "Domingo";
    else if (diaSemana == 2)
        nomeDiaSemana = "Segunda-feira";
    else if (diaSemana == 3)
        nomeDiaSemana = "Terça-feira";
    else if (diaSemana == 4)
        nomeDiaSemana = "Quarta-feira";
    else if (diaSemana == 5)
        nomeDiaSemana = "Quinta-feira";
    else if (diaSemana == 6)
        nomeDiaSemana = "Sexta-feira";
    else if (diaSemana == 7)
        nomeDiaSemana = "Sábado";
    else
    {
        nomeDiaSemana = "Desconhecido...";
    }

    Console.WriteLine("Dia escolhido: " + nomeDiaSemana + "\n\rAperte uma tecla para continuar...");
    Console.ReadKey(true);
    Console.Clear();

    leitura++;
}
```



Estruturas de repetição (*for*)

A instrução *for* permite implementar uma ideia semelhante da instrução *while*, que é repetir determinadas vezes um trecho do código, porém de uma maneira mais formal.

No *for* a estrutura de inicialização, expressão lógica e atualização da variável, são escritas de uma maneira mais elegante.

Estruturas de repetição (for)

Escopo da estrutura for:

```
for(INICIALIZAÇÃO; EXPRESSÃO LÓGICA; ATUALIZAÇÃO DA VAR.)  
{  
    BLOCO DE INSTRUÇÕES  
}
```

ou:

```
for(INICIALIZAÇÃO; EXPRESSÃO LÓGICA; ATUALIZAÇÃO DA VAR)  
    LINHA DE INSTRUÇÃO;
```



Estruturas de repetição (*for*)

Apontamentos para a instrução *for*:

- Você pode omitir qualquer uma das três partes da instrução (“inicialização”, expressão lógica e/ou atualização da variável de controle);

Estruturas de repetição (*for*)

Se você “omitir” a inicialização:

- Não conseguirá compilar, exceto se declarar e inicializar a variável fora do escopo do *for*.

```
int i = 0;  
for (; i < 10; i++)  
{  
    BLOCO DE INSTRUÇÕES  
}
```

Estruturas de repetição (*for*)

Se você omitir a expressão lógica:

- A expressão será considerada como *true*.
Gerando uma repetição eterna...

```
for (int i = 0; ; i++)  
{  
    BLOCO DE INSTRUÇÕES  
}
```

Estruturas de repetição (*for*)

Se você omitir a atualização da variável de controle:

- A variável não irá atualizar, logo a o resultado da expressão será sempre o mesmo.
 - Caso seja *true*, fará uma repetição eterna.

```
for (int i = 0; i < 10; )  
{  
    BLOCO DE INSTRUÇÕES  
}
```


Estruturas de repetição (*for*)

Se você omitir a atualização da variável de controle:

- A variável não irá atualizar, logo o resultado da expressão será sempre o mesmo.
 - Caso seja *false*, NÃO executará o bloco de instruções.

```
for (int i = 10; i < 10; )  
{  
    BLOCO DE INSTRUÇÕES  
}
```

Estruturas de repetição (*for*)

Se você “omitir” a inicialização e omitir a atualização da variável de controle:

- Poderá fazer um *while* de forma diferente!

```
int i = 0;
for ( ; i < 10; )
{
    BLOCO DE INSTRUÇÕES
    ATUALIZAR VARIÁVEL DE CONTROLE
}
```



Estruturas de repetição (*for*)

Se você omitir as três partes:

- Obterá uma repetição infinita. Afinal, a expressão lógica será considerada *true* por padrão.

```
for ( ; ; )  
{  
    BLOCO DE INSTRUÇÕES  
}
```

*Exercício a seguir...

Exercício Prático (*for*)

Exercício 2a:

Name: estrutura_for_dez

Solution: Add to solution

Copie o código do exercício anterior e altere a repetição para ser feita por 10 vezes, utilizando a instrução *for*. Mostre em que passo a leitura se encontra.

1 = Domingo ... 7 = Sábado

- O programa só finaliza depois de ler 10 vezes.
- As mensagens (nomes dos dias) devem ser exibidas após cada leitura (normalmente como era antes!).

Resposta - Exercício Prático (for)

```
string nomeDiaSemana = "";
int leitura = 1;

for(int i = 1; i <= 10; i++)
{
    Console.WriteLine("Programa de estrutura while - Leitura: " + i);
    Console.WriteLine("Digite um numero inteiro entre 1 e 7");
    int diaSemana = Int32.Parse(Console.ReadLine());

    if (diaSemana == 1)
        nomeDiaSemana = "Domingo";
    else if (diaSemana == 2)
        nomeDiaSemana = "Segunda-feira";
    else if (diaSemana == 3)
        nomeDiaSemana = "Terça-feira";
    else if (diaSemana == 4)
        nomeDiaSemana = "Quarta-feira";
    else if (diaSemana == 5)
        nomeDiaSemana = "Quinta-feira";
    else if (diaSemana == 6)
        nomeDiaSemana = "Sexta-feira";
    else if (diaSemana == 7)
        nomeDiaSemana = "Sábado";
    else
    {
        nomeDiaSemana = "Desconhecido...";
    }

    Console.WriteLine("Dia escolhido: " + nomeDiaSemana + "\n\rAperte uma tecla para continuar...");
    Console.ReadKey(true);
    Console.Clear();
}
```

Exercício Prático (for)

Exercício 2b:

Name: estrutura_for_break

Solution name: Add to solution

Copie o código do exercício anterior e altere-o para repetir o preenchimento enquanto o usuário colocar um dado válido (dentro do intervalo 1-7), caso contrário, escreva “Desconhecido” e encerre o programa (deixem o “PAUSE” antes de encerrar.

Utilize o *for* omitindo as três partes!

1 = Domingo ... 7 = Sábado

PS: Utilize a instrução *break*;

Resposta - Exercício Prático (for)

```
string nomeDiaSemana = "";
for ( ; ; )
{
    Console.WriteLine("Programa de estrutura while - Leitura: ");
    Console.WriteLine("Digite um numero inteiro entre 1 e 7");
    int diaSemana = Int32.Parse(Console.ReadLine());

    if (diaSemana == 1)
        nomeDiaSemana = "Domingo";
    else if (diaSemana == 2)
        nomeDiaSemana = "Segunda-feira";
    else if (diaSemana == 3)
        nomeDiaSemana = "Terça-feira";
    else if (diaSemana == 4)
        nomeDiaSemana = "Quarta-feira";
    else if (diaSemana == 5)
        nomeDiaSemana = "Quinta-feira";
    else if (diaSemana == 6)
        nomeDiaSemana = "Sexta-feira";
    else if (diaSemana == 7)
        nomeDiaSemana = "Sábado";
    else
    {
        nomeDiaSemana = "Desconhecido!";
        break;
    }

    Console.WriteLine("Dia escolhido: " + nomeDiaSemana + "\n\rAperte uma tecla para continuar...");
    Console.ReadKey(true);
    Console.Clear();
}

Console.WriteLine("Dia escolhido: " + nomeDiaSemana + "\n\rAperte uma tecla para continuar...");
Console.ReadKey(true);
```

Estruturas de repetição (*do-while*)

A instrução *do-while* também implementa a ideia de repetir um determinado bloco de instruções (bem parecido com o *while*). Porém, uma pequena diferença deve ser salientada: o pós-teste.

No *while* o bloco de instruções é primeiramente executado, para depois ser testada a expressão lógica.

Estruturas de repetição (*do-while*)

Escopo da instrução *do-while*:

```
do  
{
```

BLOCO DE INSTRUÇÕES

```
}  
while (EXPRESSÃO LÓGICA);
```

Exercício Prático (do-while)

Exercício 3:

Name: estrutura_do-while

Solution name: Add to solution

Utilize a *do-while* e, com base no exercício anterior, mantenha a lógica de repetição do preenchimento enquanto o usuário colocar um dado válido (dentro do intervalo 1-7), caso contrário, escreva “Desconhecido” e encerre o programa (deixem o “PAUSE” antes de encerrar).

1 = Domingo ... 7 = Sábado

Resposta - Exercício Prático (do-while)

```
string nomeDiaSemana = "";
int diaSemana = 0;
do
{
    Console.WriteLine("Programa de estrutura while - Leitura: ");
    Console.WriteLine("Digite um numero inteiro entre 1 e 7");
    diaSemana = Int32.Parse(Console.ReadLine());

    if (diaSemana == 1)
        nomeDiaSemana = "Domingo";
    else if (diaSemana == 2)
        nomeDiaSemana = "Segunda-feira";
    else if (diaSemana == 3)
        nomeDiaSemana = "Terça-feira";
    else if (diaSemana == 4)
        nomeDiaSemana = "Quarta-feira";
    else if (diaSemana == 5)
        nomeDiaSemana = "Quinta-feira";
    else if (diaSemana == 6)
        nomeDiaSemana = "Sexta-feira";
    else if (diaSemana == 7)
        nomeDiaSemana = "Sábado";
    else
    {
        nomeDiaSemana = "Desconhecido!";
    }

    Console.WriteLine("Dia escolhido: " + nomeDiaSemana + "\n\rAperte uma tecla para continuar...");
    Console.ReadKey(true);
    Console.Clear();
}
while ( (diaSemana > 0) && (diaSemana < 8) );
```



Resumo de aula

- Nesta aula vimos:
 - Estruturas de repetição
 - while;
 - Aplicação; Escopo; Exercício prático.
 - for;
 - Aplicação; Escopo; Exercício prático.
 - do-while;
 - Aplicação; Escopo; Exercício prático.

Resumo de aula

- Mandem as anotações de aula para a tarefa no moodle.
- A tarefa “Trabalho – 03” é correspondente ao trabalho sobre Arquitetura do .NET