

Programação orientada a objetos

THIAGO ALVES DE MOURA LIMA

Prof: ROSENCLEVER LOPES GAZONI

Selecione a opção que mostra como podemos criar e definir o valor de uma variável, para poder representar a quantidade de dias de um mês. A. `int diasDoMes; diasDoMes = 30;` B. `diasDoMes; diasDoMes = 30;` C. `int diasDoMes diasDoMes = 30` D. `int Dias; Dias => 31;`

A opção correta para criar e definir o valor de uma variável que representa a quantidade de dias de um mês em C# é:

A.

```
int diasDoMes;  
diasDoMes = 30;
```

Esta opção mostra a declaração da variável `diasDoMes` do tipo `int` e, em seguida, atribui o valor 30 a ela, que pode representar, por exemplo, o número de dias em abril, junho, setembro ou novembro.

Mão na massa 1. Dentro do seu projeto "ConceitosBasicos", altere a classe Program para: a) Criar uma variável com nome `idade`, do tipo `int` e atribua um valor. b) Imprimir a variável `idade`.

```
using System;  
  
namespace ConceitosBasicos  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            // a) Criar uma variável com nome idade, do tipo int e  
            atribuir um valor.  
            int idade = 25; // Você pode substituir 25 pelo valor  
            desejado.  
  
            // b) Imprimir a variável idade.  
            Console.WriteLine("A idade é: " + idade);  
        }  
    }  
}
```

}

Aprendemos que é possível fazer operações entre números. Podemos até fazer operações aritméticas com variáveis de tipos diferentes, como multiplicar um inteiro por um número de ponto flutuante (int e double). Verificando cada instrução abaixo diga qual das opções está errada: A. double peso = 4.0; int quantidade = 10; Console.WriteLine(peso * quantidade); Deve imprimir 40 B. int brindes = 15; int pessoas = 2; Console.WriteLine(brindes / pessoas); Deve imprimir 7.5 C. int dia = 4; int outroDia = 4 + dia; Console.WriteLine(outroDia); Deve imprimir 8 D. double preco = 5.5; int ingressos = 4; Console.WriteLine(preco * ingressos); Deve imprimir 22

A opção que está **errada** é a **B**:

```
int brindes = 15;
int pessoas = 2;
Console.WriteLine(brindes / pessoas);
.
```

Deve imprimir 7.5, mas isso está incorreto porque a operação está dividindo dois inteiros, o que resultará em uma divisão inteira. Em C#, quando você divide dois inteiros, o resultado é também um inteiro, e qualquer fração é descartada. Portanto, o resultado impresso será 7, não 7.5.

O código apresentado a seguir não compila. • Qual o motivo? • Como resolver sem alterar o tipo de dado das variáveis? double salario = 1270.50; int valor = salario;

O motivo pelo qual o código não compila é porque está tentando atribuir diretamente um valor do tipo double a uma variável do tipo int. Em C#, isso é chamado de “conversão implícita” e não é permitido de um tipo de ponto flutuante para um tipo inteiro, pois pode haver perda de precisão.

```
double salario = 1270.50;
int valor = (int)salario;
.
```

Indique todas as alternativas abaixo que mostrem o resultado "A idade de Marcos é 30!":
a) int idade = 30; console.log.writeline("A idade de marcos é " + idade + "!"); b)double

idade = 30.0; console.log.writeline("A idade de marcos é " +(int) idade + "!") ; c)int idade = 20; console.log.writeline("A idade de marcos é 30!") ; d)int idade = 3 * 10 console.log.writeline("A idade de marcos é " + idade + "!") ; e)int idade = 10 + 20; console.log.writeline("A idade de marcos é " + trinta) ;

As alternativas que mostram o resultado “A idade de Marcos é 30!” são:

a)

```
int idade = 30;  
Console.WriteLine("A idade de Marcos é " + idade + "!");
```

b)

```
double idade = 30.0;  
Console.WriteLine("A idade de Marcos é " + (int)idade + "!");
```

d)

```
int idade = 3 * 10;  
Console.WriteLine("A idade de Marcos é " + idade + "!");
```

e)

```
int idade = 10 + 20;  
Console.WriteLine("A idade de Marcos é " + idade + "!");
```

Guilherme, vendo que romulo continuava com algumas dificuldades com C#, resolveu elaborar um desafio para seu amigo concatenação de Strings, vistos nesse capítulo. O desafio continha o seguinte trecho de código: string saudacao = "Olá, meu nome é "; string nome = "Romulo "; string continuação = "e minha idade é "; int idade = 100; console.Writeline(saudação + nome + continuação + idade) ; oCódigo de Guilherme para Romulo possui algum problema? selecione uma alternativa a) Sim, Pois não é possível concentrar uma string com um int b) Não há problemas a concentração pode ser feita c) Sim, pois é correto é Int e não int

A alternativa correta é:

b) Não há problemas, a concatenação pode ser feita

Romulo , depois de assistir os videos deste capitulo, resolveu testar o que foi dito em aula e fez um codigo pra somar dois valores quaisquer. `string parcela_1 = "10"; string parcela_2 = "20"; Console.WriteLine(parcela_1 + parcela_2);` Romulo terá o resultado esperado (30) com este trecho de codigo? selicione uma alternativa a) Não, pois não podemos utilizar o sinal + neste caso. b) Sim, porem o resultado sera "30" e não 30. c) Não o resulatato será 1020. d) Sim. o reusultado sera 30

A alternativa correta é:

c) Não, o resultado será 1020.

Qual será o resultado da compilação do codigo apresentado ? `static void Main (String args[]) { bool foiPromovido = true; if(foiPromovido) { double salrio = 4200.0; } else { double salario = 3800.0; } Console.WriteLine(salario); }`

a) o codigo compila mas apresenta um erro de execução, pois a variavel salario não foi inicializada. b) O codigo não bem compila pois declaramos a variavel salario duas vezes c) O codigo nem compila pois fora do bloco if/else a variavel salario nao existe mais

A alternativa correta é:

c) O código nem compila, pois fora do bloco if/else a variável salario não existe mais.

Dentro da sua pasta de projetos crie uma subpasta chamada 1bim • Dentro da pasta CSharp1bim, crie um novo projeto chamado fixacaofundamentos, que estará dentro da solução conceitosBasicosCSharp • Resolva os exercícios a seguir, sendo para cada um deles crie um novo arquivo cujo nome deve ser relacionado ao seu conteúdo, por exemplo, para o exercício 1, o nome deste arquivo deve ser SomaDoisNumeros (sempre respeitando as mesmas regras

de nomenclatura utilizadas para variáveis, exceto pelo fato de iniciarem em maiúsculas): 1. Crie um programa que peça dois números inteiros e ao final imprima a soma dos números fornecidos. (SomaDoisNumeros)

```
using System;

namespace fixacaofundamentos
{
    class SomaDoisNumeros
    {
        static void Main(string[] args)
        {
            Console.Write("Digite o primeiro número inteiro: ");
            int numero1 = Convert.ToInt32(Console.ReadLine());

            Console.Write("Digite o segundo número inteiro: ");
            int numero2 = Convert.ToInt32(Console.ReadLine());

            int soma = numero1 + numero2;
            Console.WriteLine($"A soma dos números fornecidos é:
{soma}");
        }
    }
}
```

Escreva o programa MetrosMilimetros o qual deve ler um valor em metros e o exibi-lo em milímetros. Crie o programa CalculaAumento que deve solicitar o valor atual do salário e o percentual de aumento. Ao final deverá exibir todas as informações: salário inicial, percentual de aumento, valor do aumento e valor do novo salário. Crie o programa CalculaDesconto que deve solicitar o valor atual de uma mercadoria e o percentual de desconto. Ao final deverá exibir todas as informações: valor atual, percentual de desconto, valor do desconto e valor final da mercadoria. Escreva o programa AluguelCarro, que deverá informar o valor a ser pago pelo aluguel de um veículo, sabendo que o valor da diária de aluguel do veículo é R\$95,00 e o valor por quilômetro percorrido é R\$0,35. Os dados de entrada são: Total de dias alugados, quilometragem inicial e quilometragem final.

Programa MetrosMilímetros:

```
using System;

class MetrosMilímetros
{
    static void Main()
    {
        Console.Write("Digite o valor em metros: ");
        double metros = double.Parse(Console.ReadLine());
        double milímetros = metros * 1000;
        Console.WriteLine($"{metros} metros é igual a {milímetros} milímetros.");
    }
}
```

Programa CalculaAumento:

```
using System;

class CalculaAumento
{
    static void Main()
    {
        Console.Write("Digite o salário atual: R$");
        double salarioAtual = double.Parse(Console.ReadLine());
        Console.Write("Digite o percentual de aumento: ");
        double percentualAumento = double.Parse(Console.ReadLine()) /
100;

        double valorAumento = salarioAtual * percentualAumento;
        double novoSalario = salarioAtual + valorAumento;

        Console.WriteLine($"Salário inicial: R${salarioAtual}");
        Console.WriteLine($"Percentual de aumento: {percentualAumento *
100}%");
        Console.WriteLine($"Valor do aumento: R${valorAumento}");
        Console.WriteLine($"Novo salário: R${novoSalario}");
    }
}
```

Programa CalculaDesconto:

```
using System;

class CalculaDesconto
{
    static void Main()
    {
        Console.Write("Digite o valor atual da mercadoria: R$");
        double valorAtual = double.Parse(Console.ReadLine());
        Console.Write("Digite o percentual de desconto: ");
        double percentualDesconto = double.Parse(Console.ReadLine()) /
100;
        double valorDesconto = valorAtual * percentualDesconto;
        double valorFinal = valorAtual - valorDesconto;

        Console.WriteLine($"Valor atual: R${valorAtual}");
        Console.WriteLine($"Percentual de desconto: {percentualDesconto *
100}%");
        Console.WriteLine($"Valor do desconto: R${valorDesconto}");
        Console.WriteLine($"Valor final da mercadoria: R${valorFinal}");
    }
}
```

Programa AluguelCarro:

```
using System;

class AluguelCarro
{
    static void Main()
    {
        const double valorDiaria = 95.00;
        const double valorKm = 0.35;

        Console.Write("Total de dias alugados: ");
        int totalDias = int.Parse(Console.ReadLine());
        Console.Write("Quilometragem inicial: ");
        double kmInicial = double.Parse(Console.ReadLine());
        Console.Write("Quilometragem final: ");
        double kmFinal = double.Parse(Console.ReadLine());
    }
}
```

```
double totalKm = kmFinal - kmInicial;
double custoAluguel = (totalDias * valorDiaria) + (totalKm *
valorKm);

    Console.WriteLine($"O valor a ser pago pelo aluguel do veículo é:
R${custoAluguel}");
}
}
```

Crie um programa que solicite o peso, a altura e o gênero do usuário. Ao final, deve ser informado o IMC e a classificação na qual ele se encontra de acordo com a tabela a seguir: Imc Classificação abaixo de 18,5 abaixo do peso entre 18,6 e 24,9 peso ideal (parabens) entre 25,0 e 29,0 Levemente acima do peso entre 30,0 e 34,9 Obesidade grau II entre 35,0 e 39,9 Obesidade grau II (severa) acima de 40 Obesidade grau III (morbida)

```
using System;

class CalculoIMC
{
    static void Main()
    {
        Console.Write("Digite seu peso em quilogramas (kg): ");
        double peso = double.Parse(Console.ReadLine());

        Console.Write("Digite sua altura em metros (m): ");
        double altura = double.Parse(Console.ReadLine());

        double imc = peso / (altura * altura);
        Console.WriteLine($"Seu IMC é: {imc:F2}");

        if (imc < 18.5)
        {
            Console.WriteLine("Classificação: abaixo do peso");
        }
        else if (imc >= 18.5 && imc <= 24.9)
        {
            Console.WriteLine("Classificação: peso ideal (parabéns)");
        }
        else if (imc >= 25.0 && imc <= 29.9)
        {
            Console.WriteLine("Classificação: Levemente acima do peso");
        }
        else if (imc >= 30.0 && imc <= 34.9)
        {
            Console.WriteLine("Classificação: Obesidade grau I");
        }
        else if (imc >= 35.0 && imc <= 39.9)
        {
            Console.WriteLine("Classificação: Obesidade grau II (severa)");
        }
        else if (imc >= 40.0)
        {
            Console.WriteLine("Classificação: Obesidade grau III (morbida)");
        }
    }
}
```



```
{
    Console.WriteLine("Classificação: levemente acima do peso");
}
else if (imc >= 30.0 && imc <= 34.9)
{
    Console.WriteLine("Classificação: obesidade grau I");
}
else if (imc >= 35.0 && imc <= 39.9)
{
    Console.WriteLine("Classificação: obesidade grau II
(severa)");
}
else // imc >= 40
{
    Console.WriteLine("Classificação: obesidade grau III
(mórbida)");
}
}
}
```

Qual será a saída obtida com este código: `int contador = 1 while (contador <= 10) { cosole.WriteLine(contador); }`

```
int contador = 1;
while (contador <= 10)
{
    Console.WriteLine(contador);
    contador++;
}
```

Note que também adicionei `contador++` para incrementar o valor de `contador` a cada iteração do loop `while`, pois sem isso, o loop se tornaria infinito.

Com as correções feitas, a saída do código seria a impressão dos números de 1 a 10, cada um em uma nova linha. Aqui está a saída esperada:

```
1
2
3
4
5
6
```

7
8
9
10

Faça a transição do código apresentado para estrutura `for`: `int contador = 1;`
`while (contador <= 10) { Console.WriteLine(contador); contador++; }`

```
for (int contador = 1; contador <= 10; contador++)  
{  
    Console.WriteLine(contador);  
}
```

Clarice está em dúvida sobre o modo como o `break` funciona quando utilizado dentro de laços de repetição aninhados. Escolha a opção que descreve exatamente o funcionamento desse comando nessas situações. a) Finaliza a execução do programa no momento em que é chamado b) Para a execução do laço mais interno que contém o comando `break` e continua executando os laços mais externos. c) Para a execução de todos os laços de repetição aninhados que o contêm e continua normalmente o programa

b) Para a execução do laço mais interno que contém o comando `break` e continua executando os laços mais externos.

Fixação: Crie um programa que seja capaz de gerar a seguinte saída: 1 12 123
1234 12345

```
using System;  
  
class SequenciaNumerica  
{
```

```
static void Main()
{
    for (int i = 1; i <= 5; i++)
    {
        for (int j = 1; j <= i; j++)
        {
            Console.Write(j);
        }
        Console.WriteLine();
    }
}
```

Quando você executar este programa, ele produzirá a seguinte saída:

```
1
12
123
1234
12345
```

Crie um programa para gerar a tabuada de cada número. Crie um programa para imprimir todos os múltiplos de 3 entre 0 e 100. Crie um programa que imprima os fatoriais de 1 até 10. o fatorial de 0 é 1 o fatorial de 1 é $(0!) * 1=1$ o fatorial de 2 é $(1!) * 2=2$ o fatorial de 3 é $(2!) * 3=6$ o fatorial de 4 é $(3!) * 4=24$ o fatorial de 1 de um numero n é $n * n-1 * n-2$ ou seja o fatorial de $4! = 1 \times 2 \times 3 \times 4 = 24$ o fatorial de $6! = 1 \times 2 \times 3 \times 4 \times 5 \times 6 = 720$

```
using System;

class Tabuada
{
    static void Main()
    {
        for (int numero = 1; numero <= 10; numero++)
        {
            Console.WriteLine($"Tabuada do {numero}:");
            for (int i = 1; i <= 10; i++)
            {
                Console.WriteLine($"{numero} x {i} = {numero * i}");
            }
            Console.WriteLine(); // Espaço entre as tabuadas
        }
    }
}
```

```
    }  
  }  
}
```

Programa para imprimir todos os múltiplos de 3 entre 0 e 100:

```
using System;  
  
class MultiplosDeTres  
{  
    static void Main()  
    {  
        for (int i = 0; i <= 100; i += 3)  
        {  
            Console.WriteLine(i);  
        }  
    }  
}
```

Programa que imprime os fatoriais de 1 até 10:

```
using System;  
  
class Fatorial  
{  
    static void Main()  
    {  
        for (int n = 1; n <= 10; n++)  
        {  
            int fatorial = 1;  
            for (int i = 1; i <= n; i++)  
            {  
                fatorial *= i;  
            }  
            Console.WriteLine($"{n} fatorial de {n} é {fatorial}");  
        }  
    }  
}
```

Crie um programa que solicite ao usuário informar a sua renda e quantos dependentes possui e ao final imprima o valor de IRPF que será deduzido, tendo por base a tabela a seguir: tabela de incidencia mensal base de calculo (R\$) Aliquota(%) Parcela a deduzir do IRPF (R\$) até 1903,98 de 1.903,99 até 2826,65 7,5 142,80 de 2826,66 até 3751,05 15 354,80 de 3751,06 até 4664,68 22,5 636,13 Acima de 4.664,68 27,5 869,36

```
using System;

class CalculoIRPF
{
    static void Main()
    {
        Console.WriteLine("Informe sua renda mensal: R$");
        double renda = double.Parse(Console.ReadLine());

        Console.WriteLine("Informe a quantidade de dependentes: ");
        int dependentes = int.Parse(Console.ReadLine());

        // Dedução por dependente (valor hipotético)
        double deducacaoPorDependente = 189.59; // Valor hipotético, ajuste
        conforme a legislação vigente
        double deducacaoDependentes = dependentes * deducacaoPorDependente;

        // Renda ajustada após dedução por dependentes
        double rendaAjustada = renda - deducacaoDependentes;

        double aliquota = 0;
        double parcelaADeduzir = 0;

        if (rendaAjustada <= 1903.98)
        {
            aliquota = 0;
            parcelaADeduzir = 0;
        }
        else if (rendaAjustada <= 2826.65)
        {
            aliquota = 0.075;
            parcelaADeduzir = 142.80;
        }
        else if (rendaAjustada <= 3751.05)
        {
            aliquota = 0.15;
```

```
        parcelaADeduzir = 354.80;
    }
    else if (rendaAjustada <= 4664.68)
    {
        aliquota = 0.225;
        parcelaADeduzir = 636.13;
    }
    else
    {
        aliquota = 0.275;
        parcelaADeduzir = 869.36;
    }

    double irpf = (rendaAjustada * aliquota) - parcelaADeduzir;

    Console.WriteLine($"O valor do IRPF a ser deduzido é:
R${irpf:F2}");
    }
}
```

para simular um Jogo da Velha:

```
using System;

class JogoDaVelha
{
    static char[,] tabuleiro = new char[3, 3];
    static char jogadorAtual = 'X';

    static void Main()
    {
        InicializarTabuleiro();
        Jogar();
    }

    static void InicializarTabuleiro()
    {
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
```

```
        tabuleiro[i, j] = ' ';
    }
}

static void Jogar()
{
    bool jogoAtivo = true;
    int jogadas = 0;

    do
    {
        Console.Clear();
        ExibirTabuleiro();
        jogadas++;
        if (jogadas > 9)
        {
            Console.WriteLine("O jogo acabou sem vencedores.");
            break;
        }

        Console.WriteLine($"Veza do jogador {jogadorAtual}:");
        Console.WriteLine("Escolha a linha (0, 1, 2): ");
        int linha = int.Parse(Console.ReadLine());
        Console.WriteLine("Escolha a coluna (0, 1, 2): ");
        int coluna = int.Parse(Console.ReadLine());

        if (tabuleiro[linha, coluna] != ' ')
        {
            Console.WriteLine("Esta posição já está ocupada. Pressione qualquer tecla para tentar novamente.");
            Console.ReadKey();
            continue;
        }

        tabuleiro[linha, coluna] = jogadorAtual;
        jogadorAtual = jogadorAtual == 'X' ? 'O' : 'X';

        jogoAtivo = !VerificarVencedor();

    } while (jogoAtivo);
}
```

```
static void ExibirTabuleiro()
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            Console.Write(tabuleiro[i, j]);
            if (j < 2) Console.Write("|");
        }
        Console.WriteLine();
        if (i < 2) Console.WriteLine("-----");
    }
}

static bool VerificarVencedor()
{
    for (int i = 0; i < 3; i++)
    {
        // Verifica linhas
        if (tabuleiro[i, 0] == tabuleiro[i, 1] && tabuleiro[i, 1] ==
tabuleiro[i, 2] && tabuleiro[i, 0] != ' ')
        {
            AnunciarVencedor(tabuleiro[i, 0]);
            return true;
        }
        // Verifica colunas
        if (tabuleiro[0, i] == tabuleiro[1, i] && tabuleiro[1, i] ==
tabuleiro[2, i] && tabuleiro[0, i] != ' ')
        {
            AnunciarVencedor(tabuleiro[0, i]);
            return true;
        }
    }
    // Verifica diagonais
    if (tabuleiro[0, 0] == tabuleiro[1, 1] && tabuleiro[1, 1] ==
tabuleiro[2, 2] && tabuleiro[0, 0] != ' ' ||
        tabuleiro[0, 2] == tabuleiro[1, 1] && tabuleiro[1, 1] ==
tabuleiro[2, 0] && tabuleiro[0, 2] != ' ')
    {
        AnunciarVencedor(tabuleiro[1, 1]);
        return true;
    }
}
```



```
        return false;
    }

    static void AnunciarVencedor(char jogador)
    {
        Console.Clear();
        ExibirTabuleiro();
        Console.WriteLine($"Jogador {jogador} venceu!");
    }
}
```

1. Programa para adivinhar um número entre 0 e 100 com 10 tentativas:

```
using System;

class AdivinharNumero
{
    static void Main()
    {
        Random random = new Random();
        int numeroSecreto = random.Next(0, 101);
        int tentativas = 10;
        Console.WriteLine("Você tem 10 chances para adivinhar um número inteiro entre 0 e 100.");

        for (int tentativa = 1; tentativa <= tentativas; tentativa++)
        {
            Console.Write($"Tentativa {tentativa}/{tentativas}: Qual é o seu palpite? ");
            int palpite = int.Parse(Console.ReadLine());

            if (palpite == numeroSecreto)
            {
                Console.WriteLine($"Parabéns! Você acertou o número {numeroSecreto}!");
                return;
            }
            else if (palpite < numeroSecreto)
            {
                Console.WriteLine("Mais alto. Tente novamente.");
            }
        }
    }
}
```

```
    }  
    else  
    {  
        Console.WriteLine("Mais baixo. Tente novamente.");  
    }  
}  
Console.WriteLine($"Suas chances acabaram. O número era  
{numeroSecreto}.");  
}  
}
```

2. Programa para calcular o valor total de um veículo financiado:

```
using System;  
  
class CalculoFinanciamento  
{  
    static void Main()  
    {  
        Console.Write("Informe o valor do veículo: R$ ");  
        double valorVeiculo = double.Parse(Console.ReadLine());  
        Console.Write("Informe o total de parcelas para pagamento: ");  
        int parcelas = int.Parse(Console.ReadLine());  
        Console.Write("Informe a taxa mensal por parcela (em %): ");  
        double taxaMensal = double.Parse(Console.ReadLine());  
  
        double valorTotalPago = valorVeiculo * Math.Pow(1 + (taxaMensal /  
100), parcelas);  
  
        Console.WriteLine($"O valor total pago no veículo será de: R$  
{valorTotalPago:F2}");  
    }  
}
```

3. Programa para calcular o montante acumulado para aposentadoria e a renda mensal:

```
using System;
```

```
class CalculoAposentadoria
{
    static void Main()
    {
        Console.Write("Informe sua idade atual: ");
        int idadeAtual = int.Parse(Console.ReadLine());
        Console.Write("Com quantos anos você pretende se aposentar? ");
        int idadeAposentadoria = int.Parse(Console.ReadLine());
        Console.Write("Qual valor você pretende guardar mensalmente? R$");
        double valorMensal = double.Parse(Console.ReadLine());
        Console.Write("Qual a taxa de rendimento mensal da aplicação (em %)? ");
        double taxaRendimento = double.Parse(Console.ReadLine());

        int anosContribuicao = idadeAposentadoria - idadeAtual;
        int mesesContribuicao = anosContribuicao * 12;
        double montanteAcumulado = 0;

        for (int mes = 1; mes <= mesesContribuicao; mes++)
        {
            montanteAcumulado = (montanteAcumulado + valorMensal) * (1 + taxaRendimento / 100);
        }

        Console.WriteLine($"O montante acumulado até a aposentadoria será de: R$ {montanteAcumulado:F2}");

        double rendaMensal = montanteAcumulado * (taxaRendimento / 100);
        Console.WriteLine($"Mantendo-se o valor na mesma aplicação, sua renda mensal seria de: R$ {rendaMensal:F2}");
    }
}
```