**Documentação do Código da Calculadora**

**Visão Geral**

Este case de projeto implementa uma calculadora simples que processa uma fila de operações aritméticas. As operações são realizadas sequencialmente, e os resultados são armazenados em uma pilha para serem exibidos ao final. A calculadora suporta as operações de adição, subtração, multiplicação e divisão.

**Estrutura do Código**

O código está organizado em três namespaces principais:

1. **Calculadora**: Contém a classe **Operacoes** que define a estrutura de uma operação aritmética.
2. **Calculadora**: Contém a classe **Calculadora** que realiza os cálculos.
3. **Calculadora**: Contém a classe **Program** que gerencia a fila de operações e a execução do programa.
4. [Capture a atenção do leitor com uma ótima citação do documento ou use este espaço para enfatizar um ponto-chave. Para colocar essa caixa de texto em qualquer lugar na página, basta arrastá-la.]

**Classes e Métodos**

**Classe Operacoes**

Esta classe define a estrutura de uma operação aritmética.

1 namespace Calculadora {

2 public class Operacoes {

3 public long valorA { get; set; } // Primeiro operando

4 public char operador { get; set; } // Operador aritmético (+, -, \*, /)

5 public int valorB { get; set; } // Segundo operando

6 public long resultado { get; set; } // Resultado da operação

7 }

8}

**Classe Calculadora**

Esta classe contém os métodos para realizar os cálculos aritméticos.

1 namespace Calculadora {

2 public class Calculadora {

3 public Operacoes calcular(Operacoes operacao) {

4 switch (operacao.operador) {

5 case '+':

6 operacao.resultado = soma(operacao);

7 break;

8 case '-':

9 operacao.resultado = subtracao(operacao);

10 break;

11 case '\*':

12 operacao.resultado = multiplicacao(operacao);

13 break;

14 case '/':

15 operacao.resultado = divisao(operacao);

16 break;

17 default:

18 operacao.resultado = 0;

19 break;

20 }

21 return operacao;

22 }

23

24 public long soma(Operacoes operacao) {

25 return operacao.valorA + operacao.valorB;

26 }

27

28 public long subtracao(Operacoes operacao) {

29 return operacao.valorA - operacao.valorB;

30 }

31

32 public long multiplicacao(Operacoes operacao) {

33 return operacao.valorA \* operacao.valorB;

34 }

35

36 public long divisao(Operacoes operacao) {

return operacao.valorA / operacao.valorB;

}

}

}

**Classe Program**

Esta classe gerencia a fila de operações e a execução do programa

1 using System;

2using System.Collections;

3using System.Collections.Generic;

4

5namespace Calculadora {

6 class Program {

7 static void Main(string[] args) {

8 Queue<Operacoes> filaOperacoes = new Queue<Operacoes>();

9 filaOperacoes.Enqueue(new Operacoes { valorA = 2, valorB = 3, operador = '+' });

10 filaOperacoes.Enqueue(new Operacoes { valorA = 14, valorB = 8, operador = '-' });

11 filaOperacoes.Enqueue(new Operacoes { valorA = 5, valorB = 6, operador = '\*' });

12 filaOperacoes.Enqueue(new Operacoes { valorA = 2147483647, valorB = 2, operador = '+' });

13 filaOperacoes.Enqueue(new Operacoes { valorA = 18, valorB = 3, operador = '/' });

14

15 Calculadora calculadora = new Calculadora();

16 Stack<long> resultados = new Stack<long>();

17

18 while (filaOperacoes.Count > 0) {

19 Operacoes operacao = filaOperacoes.Dequeue();

20 calculadora.calcular(operacao);

21 Console.WriteLine("{0} {1} {2} = {3}", operacao.valorA, operacao.operador, operacao.valorB, operacao.resultado);

22 resultados.Push(operacao.resultado);

23 ImprimirOperacoesRestantes(filaOperacoes);

24 }

25

26 ImprimirResultados(resultados);

27 }

28

29 static void ImprimirOperacoesRestantes(Queue<Operacoes> filaOperacoes) {

30 Console.WriteLine("Operações restantes na fila:");

31 foreach (var operacao in filaOperacoes) {

32 Console.WriteLine("{0} {1} {2}", operacao.valorA, operacao.operador, operacao.valorB);

33 }

34 Console.WriteLine();

35 }

36

37 static void ImprimirResultados(Stack<long> resultados) {

38 Console.WriteLine("Resultados dos cálculos:");

39 while (resultados.Count > 0) {

40 Console.WriteLine(resultados.Pop());

41 }

42 }

43 }

44}

**Explicação do Código**

**Classe Operacoes**

* **valorA**: Primeiro operando da operação aritmética. Tipo long para suportar valores grandes.
* **operador**: Operador aritmético (+, -, \*, /).
* **valorB**: Segundo operando da operação aritmética. Tipo int.
* **resultado**: Resultado da operação aritmética. Tipo long.

**Classe Calculadora**

* **calcular**: Método principal que recebe uma operação e chama o método correspondente com base no operador.
* **soma**: Realiza a adição dos operandos.
* **subtracao**: Realiza a subtração dos operandos.
* **multiplicacao**: Realiza a multiplicação dos operandos.
* **divisao**: Realiza a divisão dos operandos.

**Classe Program**

* **Main**: Método principal que inicializa a fila de operações, processa cada operação e armazena os resultados em uma pilha.
* **ImprimirOperacoesRestantes**: Método auxiliar que imprime as operações restantes na fila.
* **ImprimirResultados**: Método auxiliar que imprime os resultados armazenados na pilha.

**Execução do Programa**

1. **Inicialização da Fila de Operações**: As operações são enfileiradas na filaOperacoes.
2. **Processamento das Operações**: Cada operação é retirada da fila, processada pela calculadora, e o resultado é armazenado na pilha resultados.
3. **Impressão das Operações Restantes**: Após cada cálculo, as operações restantes na fila são impressas.
4. **Impressão dos Resultados**: Após todas as operações serem processadas, os resultados são desempilhados e impressos.

**Exemplo de Saída**

1 2 + 3 = 5

2 Operações restantes na fila:

3 14 - 8

4 5 \* 6

5 2147483647 + 2

6 18 / 3

7

8 14 - 8 = 6

9 Operações restantes na fila:

10 5 \* 6

11 2147483647 + 2

12 18 / 3

13

14 5 \* 6 = 30

15 Operações restantes na fila:

16 2147483647 + 2

17 18 / 3

18

19 2147483647 + 2 = 2147483649

20 Operações restantes na fila:

21 18 / 3

22

23 18 / 3 = 6

24 Operações restantes na fila:

25

26 Resultados dos cálculos:

27 6

28 2147483649

29 30

30 6

31 5