

0. Presentación del Informe

Título del Informe:

Tarea Obligatoria Nro. 1: Operaciones.

Autor:

Rodrigo Ciosek.

Materia / Asignatura:

Algoritmos y Estructuras de Datos 1.

Profesor:

Carlos Mascheroni.

Fecha de entrega:

8 de abril de 2025

1. Introducción

Desarrollar una función que, dada cierta operación entre dos valores devuelva el resultado correspondiente utilizando únicamente sumas y restas. No está permitido el uso de operadores matemáticos directos distintos de + y – (adición y sustracción).

La operación por realizar se indicará por medio de un tercer parámetro. Si los valores no son adecuados para la operación solicitada, la función deberá manejar dicha situación de forma apropiada.

2. Análisis

2.1 Definición y entendimiento del problema

Crear una función o procedimiento que retorne el resultado de operaciones aritméticas utilizando solamente la adición y la sustracción. Se pasarán tres valores como argumentos, dos valores enteros y un carácter que define la operación aritmética.

2.2 Entradas y salidas:

2.2.1 Entradas:

- Un **valor entero** “a”.
- Un **valor entero** “b”.
- Un carácter “op” que represente un símbolo aritmético. Puede ser '+', '-', '*', '/' .

2.2.2 Salidas:

- Un numero flotante que representa el resultado de una operación aritmética con los valores **a**, **b** y el **operador**.

2.3 Pre y Post condiciones

2.3.1 Pre condiciones

- "a", "b", "op", no sea nulo ni vacío
- Si "op" = 0, "num1" o "num2" != 0
- op == + o op == - o op == + o op == +

2.3.2 Post condiciones

- Retorna un numero entero o flotante como resultado de la operación aritmética entre a y b.
- Si alguno de los parámetros está vacío devolverá un mensaje de error.
- Si se intenta realizar una división por 0 devolverá un mensaje de error.

3. Diseño

3.2 Pseudocódigo

Clase Algoritmo.java

Funcion Calculadora(a, op, b) : Flotante

Si op != '-' Y op != '+' Y op != '/' Y op != '*'

Retornar "NaN"

FinSi

Segun op Hacer

Caso '-':

Retornar Utilidades.Resta(a,b)

Caso '+':

Retornar Utilidades.Suma(a,b)

Caso '/':

Retornar Utilidades.Division(a,b)

Caso '*':

Retornar Utilidades.Multiplicacion(a,b)

FinFuncion

Clase Utilidades.Java

Funcion suma(a,b) : Entero

Retornar a + b

FinFuncion

Funcion resta(a,b) : Entero

Retornar a - b

FinFuncion

Funcion multiplicacion(a,b) : Entero

resultado = 0

absA = FuncionValorAbsoluto(a)

absB = FuncionValorAbsoluto(b)

Para i = 0 Hasta i=absB Hacer

resultado = resultado + absA;

FinPara

Si a < 0 Y b < 0

Retornar FuncionValorAbsoluto(resultado)

FinSi

Si a < 0 O b < 0

Retornar resultado * -1

FinSi

Retornar resultado

FinFuncion

Funcion division(a,b) : Flotante

resultado = 0

resto = 0

cociente = 0

absA = ValorAbsoluto(a)

absB = ValorAbsoluto(b)

restoAux = 0

restoDivLarga = 0

decimal = 0

// Calcular cociente

Mientras restaA >= absB Hacer

 restaA = restaA - absB

 cociente = cociente + 1

FinMientras

resto = restaA

resultado = cociente

Si resto = 0

 Retornar resultado

FinSi

// Division larga para conseguir el resto en decimal

// 1. Mutliplicar el resto * 10

restoAux = resto

Para i = 1 hasta i = 10 Hacer

 resto = resto + restoAux

FinPara

// 2. Conseguir cuantas veces entra el divisor en el resto

Mientras resto >= absA Hacer

 resto = resto - absA

 restoDivLarga = restoDivLarga + 1

FinMientras

// 3. Convertir restoDivLarga a decimal para sumarlo al resultado

Para i = 0 Hasta i = restoDivLarga

```
decimal = decimal + 0.1
```

FinPara

```
// 4. Sumarlo a resultado
```

```
resultado = resultado + decimal
```

Retornar resultado

FinFuncion

```
*/
```

4. Implementación

Algoritmo.java

```
11 public class Algoritmo { 14 usages
12     public static float Calculadora(int a, char op, int b){ 14 usages
13         if (op != '-' && op != '+' && op != '/' && op != '*') return Float.NaN;
14
15         switch (op){
16             case '-':
17                 return (Utilidades.rest(a,b));
18             case '+':
19                 return Utilidades.suma(a,b);
20             case '/':
21                 if (b == 0) return Float.NaN;
22                 return Utilidades.division(a,b);
23             case '*':
24                 return Utilidades.multiplicacion(a,b);
25         }
26
27         return Float.NaN;
28     }
29 }
```

Utilidades.java

```
1  public class Utilidades { 4 usages
2
3      public static int suma(int a, int b) { 1 usage
4          return a + b;
5      }
6
7      public static int resta(int a, int b) { 1 usage
8          return a - b;
9      }
10
11     public static int multiplicacion(int a, int b) { 1 usage
12         int resultado = 0;
13         int absA = Math.abs(a);
14         int absB = Math.abs(b);
15         for (int i = 0; i < absB; i++) {
16             resultado += absA;
17         }
18         if (a < 0 && b < 0) return Math.abs(resultado);
19         if (a < 0 || b < 0) return resultado * -1;
20         return resultado;
21     }
```

```

23     public static float division(int a, int b) { 1 usage
24         float resultado = 0;
25         int resto = 0;
26         int cociente = 0;
27         int absA = Math.abs(a);
28         int absB = Math.abs(b);
29         int restaA = absA;
30
31         // Conseguir el cociente
32         while (restaA >= absB) {
33             restaA -= absB;
34             cociente++;
35         }
36         resto = restaA;
37         resultado = cociente;
38         if (resto == 0) return resultado;
39
40         // Division Larga para conseguir el restante en decimal
41
42         // 1. Conseguir el resto * 10
43         int restoAux = resto;
44         for (int i = 1; i < 10; i++) {
45             resto += restoAux;
46         }
47         ;
48
49         // 2. Conseguir el cuantas veces entra el divisor en el resto*10
50         int restoDivLarga = 0;
51         while (resto >= absB) {
52             resto -= absB;
53             restoDivLarga++;
54         }
55
56         //3. Convertir restoDivLarga a decimal
57         float decimal = 0;
58         for (int i = 0; i < restoDivLarga; i++) {
59             decimal += 0.1f;
60         }
61
62         resultado += decimal;
63         return resultado;
64     }
65 }

```


5. Verificación y Validación

Pruebas

- **Caso básico 1:**
 - $(-1, '+', 5) \rightarrow$ Resultado esperado: 4.0
- **Caso básico 2:**
 - $(1, '+', -5) \rightarrow$ Resultado esperado: -4.0
- **Caso básico 3:**
 - $(6, '-', 5) \rightarrow$ Resultado esperado: 1.0
- **Caso básico 4:**
 - $(5, '-', 6) \rightarrow$ Resultado esperado: -1.0
- **Caso básico 5:**
 - $(-5, '-', -6) \rightarrow$ Resultado esperado: 1.0
- **Caso básico 6:**
 - $(1, '*', 5) \rightarrow$ Resultado esperado: 5.0
- **Caso básico 7:**
 - $(6, '*', -2) \rightarrow$ Resultado esperado: -12.0
- **Caso básico 8:**
 - $(0, '*', 10) \rightarrow$ Resultado esperado: 0.0
- **Caso básico 9:**
 - $(10, '*', 0) \rightarrow$ Resultado esperado: 0.0
- **Caso básico 10:**
 - $(1, '/', -5) \rightarrow$ Resultado esperado: -0.2
- **Caso básico 11:**
 - $(5, '/', 0) \rightarrow$ Resultado esperado: NaN
- **Caso básico 12:**
 - $(0, '/', 5) \rightarrow$ Resultado esperado: 0.0
- **Caso básico 13:**
 - $(0, '/', 5) \rightarrow$ Resultado esperado: NaN
- **Caso básico 14:**
 - $(0, '&', 5) \rightarrow$ Resultado esperado: NaN

Resultados:

- **Caso básico 1:** 4.0 (Correcto)
- **Caso básico 2:** -4.0 (Correcto)
- **Caso básico 3:** 1.0 (Correcto)
- **Caso básico 4:** -1.0 (Correcto)
- **Caso básico 5:** 1.0 (Correcto)
- **Caso básico 6:** 5.0 (Correcto)
- **Caso básico 7:** -12.0 (Correcto)
- **Caso básico 8:** 0.0 (Correcto)
- **Caso básico 9:** 0.0 (Correcto)
- **Caso básico 10:** -0.2 (Correcto)
- **Caso básico 11:** NaN (Correcto)
- **Caso básico 12:** 0.0 (Correcto)
- **Caso básico 13:** NaN (Correcto)
- **Caso básico 14:** NaN (Correcto)

6. Conclusiones

Buen ejercicio para poder entender como funciona la multiplicación y la división con mas profundidad y aprender a aplicarlo en código. Me resulto complicada la división, me ayudo a investigar como podría conseguir que me devolviera el resultado con resto y aprendí un concepto que se llama división larga, asique me llevo un concepto nuevo.