

Frações

1. Implemente uma classe para representar uma Fração. Uma fração é composta por um numerador(n) e um denominador(d), ambos inteiros. O denominador deve ser sempre diferente de 0. Representamos uma fração na forma $\frac{n}{d}$.

Ao criar uma fração, você deverá tomar os seguintes cuidados:

- O denominador não pode ser igual a zero. Se isso ocorrer, lance uma exceção `ValueError`.
- A fração sempre deve ser criada na forma reduzida, isto é, você deve simplificar a fração caso ela não esteja nessa forma. Por exemplo, se for feita uma tentativa de criar a fração $\frac{8}{6}$, você deve criar ela como sendo $\frac{4}{3}$. Para simplificar completamente uma fração, basta dividir ambos numerador e denominador pelo seu maior divisor comum. Para calcular o MDC podemos usar o algoritmo de Euclides. Uma implementação em Python:

```
def mdc(a: int, b: int) -> int:
    "Calcular o maximo divisor comum de a,b."
    while b != 0:
        r = a % b
        a = b
        b = r
    return a
```

Você deve implementar as quatro operações matemáticas entre frações:

- **Adição (add):** $\frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$
- **Subtração (sub):** $\frac{a}{b} - \frac{c}{d} = \frac{ad-bc}{bd}$
- **Multiplicação (mul):** $\frac{a}{b} \times \frac{c}{d} = \frac{ac}{bd}$
- **Divisão (truediv):** $\frac{a}{b} \div \frac{c}{d} = \frac{ad}{bc}$

Forneça também a implementação dos operadores relacionais, permitindo a comparação entre frações:

- **Menor (lt):** $\frac{a}{b} < \frac{c}{d}$ somente se $ad < bc$
- **Igual (eq):** Numerador e denominador devem possuir o mesmo valor (isso só vale porque as frações estão simplificadas)
- **Menor Igual (le):** Chame os métodos criados anteriormente e retorne verdadeiro em caso de um dos dois ser verdadeiro

Conversões de tipo:

- **String (str):** retorne uma string "numerador/denominador"
- **Float (float):** retorne o valor da divisão do numerador pelo denominador
- **Inteiro (int):** retorne o valor da divisão do numerador pelo denominador convertido para inteiro

Código de Testes:

```

f1 = Fracao(16, 9)
f2 = Fracao(3, 4)
f3 = Fracao(5, 25)
f4 = Fracao(16, 4)
f5 = Fracao(5, 4)

print(f1)
print(f2)
print(f3)
print(f4)
print(f5)

print("F1 + F2 =", f1 + f2)
print("F2 - F3 =", f2 - f3)
print("F3 * F4 =", f3 * f4)
print("F4 / F5 =", f4 / f5)

print("16/2 + 16/2 =", Fracao(16, 2) + Fracao(16, 2))

print("F2 == F3 =", f2 == f3)
print("F3 > F3 =", f3 > f3)
print("F4 == 4/1 =", f4 == Fracao(4, 1))

print(f1, float(f1), int(f1))
print(f2, float(f2), int(f2))
print(f3, float(f3), int(f3))
print(f4, float(f4), int(f4))
print(f5, float(f5), int(f5))

```

Exemplo de Execução:

```

16 / 9
3 / 4
1 / 5
4
5 / 4
F1 F2 = 91 / 36
F2 - F3 = 11 / 20
F3 * F4 = 4 / 5
F4 / F5 = 16 / 5
16/2 16/2 = 16
F2 == F3 = False
F3 > F3 = False
F4 == 4/1 = True
16 / 9 1.7777777777777777 1
3 / 4 0.75 0
1 / 5 0.2 0
4 4.0 4
5 / 4 1.25 1

```