

Ejercicio:

Note Title

10/28/2009

Que la clase Fraction que sirva para manipular Fracciones. Las fracciones tienen un numerador entero y un denominador entero.

Además de la función constructora (default de numerador = 0 y default de denominador = 1), crea también funciones set y get, y las siguientes funciones:

a) Función que sume 2 fracciones

$$\frac{\text{num}}{\text{den}} + \frac{\text{f.num}}{\text{f.den}} = \frac{\frac{n1}{d1} + \frac{n2}{d2}}{\frac{d2 \times n1 + d1 \times n2}{d1 \times d2}} \rightarrow \begin{array}{l} \text{numerador} \\ \text{denominador} \end{array}$$

$\text{num} = \text{f.den} \times \text{num} + \text{den} \times \text{f.num}; \text{den} = \text{den} \times \text{f.den};$

b) Función que reste 2 fracciones

$$\frac{n1}{d1} - \frac{n2}{d2} = \frac{d2 \times n1 - d1 \times n2}{d1 \times d2} \rightarrow \begin{array}{l} \text{numerador} \\ \text{denominador} \end{array}$$

c) Función que multiplique dos fracciones.

$$\frac{n1}{d1} \times \frac{n2}{d2} = \frac{n1 \times n2}{d1 \times d2} \rightarrow \begin{array}{l} \text{numerador} \\ \text{denominador} \end{array}$$

$$\begin{array}{l} \text{num} = \text{num} \times \text{f.num}; \\ \text{den} = \text{den} \times \text{f.den}; \end{array}$$

d) Función que divida 2 fracciones

$$\frac{n1}{d1} \div \frac{n2}{d2} = \frac{n1 \times d2}{d1 \times n2} \rightarrow \begin{array}{l} \text{numerador} \\ \text{denominador} \end{array}$$

e) Función que mult. una fracción por un # entero

$$\frac{n1}{d1} \times i = \frac{n1 \times i}{d1} \rightarrow \begin{array}{l} \text{numerador} \\ \text{denominador} \end{array}$$

f) Función que imprima una fracción como:

numerador / denominador

g) Función que calcule y retorne el # real equivalente a una fracción.

$$\text{Ej: } \frac{5}{8} = 0.625$$

Recuerde que una fracción no debe tener NUNCA un denominador de cero.

```

class Fraction
{
    public:
        Fraction (int = 0, int = 1);
        void setFraction (int, int);
        void setNum (int);
        void setDen (int);
        int getNum();
        int getDen();
        void addFraction (const Fraction f);
        void subtractFraction (const Fraction f);
        void multiplyFraction (const Fraction f);
        void divideFraction (const Fraction f);
        void multiplyByInt (int);
        void printFraction();
        double obtainDecimal();

    private:
        int num;
        int den;
};

```

```

Fraction::Fraction (int n, int d)
{
    setFraction (n, d);
}

void Fraction::setFraction (int n, int d)
{
    setNum (n);
    setDen (d);
}

void Fraction::setNum (int n)
{
    num = n;
}

void Fraction::setDen (int d)
{
    if (d == 0)
        den = 1;
    else
        den = d;
}

```

```
int Fraction::getNum()  
{  
    return num;  
}
```

```
int Fraction::getDen()  
{  
    return den;  
}
```

```
void Fraction::addFraction(const Fraction f)  
{  
    num = f.den * num + den * f.num;  
    den = den * f.den;  
}
```

```
void Fraction::subtractFraction(const Fraction f)  
{  
    num = f.den * num - den * f.num;  
    den = den * f.den;  
}
```

```
void Fraction::multiplyFraction(const Fraction f)  
{  
    num = num * f.num;  
    den = den * f.den;  
}
```

```
void Fraction::divideFraction(const Fraction f)  
{  
    num = num * f.den;  
    den = den * f.num;  
}
```

```
void Fraction::multiplyByInt(int n)  
{  
    num = n * num;  
}
```

```
void Fraction::printFraction()  
{  
    cout << num << '/' << den;  
}
```

```
double Fraction::obtainDecimal()  
{  
    return (double) num / (double) den;  
}
```