Implementar un programa que sume el área de un cuadrado, un círculo y un triángulo rectángulo.

Implementar una prueba unitaria que verifique que la suma del área de las figuras funciona correctamente.

Requerimiento: Cambiar el tipo de dato de los lados de las figuras de **float** a tipo de dato **Fraccion**. Hacer los cambios necesarios para que el programa siga funcionando.

RESOLUCION

Clases:

Circulo

```
#include "Fraction.h"
class Circle
{
public:
    Circle(Fraction radius) {
        this->radius = radius;
    }

    float calculateArea() {
        return PI * radius.getDecimalNumber() * radius.getDecimalNumber();
    }
private:
    Fraction radius;
    float PI = 3.1416;
};
```

Cuadrado

```
#include "Fraction.h"
class Square
{
public:
};

        Square(Fraction side) {
                this->side = side;
        }

        float calculateArea() {
                return side.getDecimalNumber() * side.getDecimalNumber();
        }

private:
        Fraction side;
};
```

Triangulo rectangulo

```
#include "Fraction.h"
class TriangleRectangle
{
public:
        TriangleRectangle(Fraction width, Fraction height) {
                this->width = width;
                this->height = height;
        }

        float calculateArea() {
                return width.getDecimalNumber() * height.getDecimalNumber() / 2.0;
        }

private:
        Fraction width, height;
};
```

Calculador de figuras

```
#include "Square.h"
#include "TriangleRectangle.h"
#include "Circle.h"

class FiguresCalculator
{
public:
    float sumAreaOfFigures(Square square, TriangleRectangle triangle, Circle circle) {
        return square.calculateArea() + triangle.calculateArea() + circle.calculateArea();
    }
};
```

Test del calculador de figuras

```
#include "FiguresCalculator.h"
#include <iostream>

using namespace std;

class FiguresCalculatorTests
{
public:
    void testCalculatorCanSumSquareTriangleAndCircleAreasCorrectly() {
        cout << "RUNNING: testCalculatorCanSumSquareTriangleAndCircleAreasCorrectly" << endl;

        // SETUP
        Fraction fraction1(40, 10);
        Square square(fraction1); // 16
        TriangleRectangle triangle(Fraction(40, 10), Fraction(60, 10)); // 12
        Circle circle(Fraction(50, 10)); // 78.54
        FiguresCalculator calculator;
        float expectedResult = 106.54;
        float receivedResult = 0.0;

        // EXECUTION
        receivedResult = calculator.sumAreaOfFigures(square, triangle, circle);

        // ASSERTION
        if (receivedResult == expectedResult) {
            cout << "La prueba paso";
        }
        else {
            cout << "La prueba fallo" << endl;
            cout << "Se esperaba: " << expectedResult << " y se recibio: " << receivedResult << endl;
        }
        // TEAR DOWN
    }
};
```

Fracción

```
class Fraction
{
public:
    Fraction() {
        this->numerator = 1;
        this->denominator = 1;
    }
    Fraction(int numerator, int denominator) {
        this->numerator = numerator;
        this->denominator = denominator;
    }

    float getDecimalNumber() {
        return numerator / denominator;
    }

private:
    int numerator;
    int denominator;
};
```

Main

```
#include <iostream>
#include "FiguresCalculatorTests.h"

int main()
{
   FiguresCalculatorTests tests;
   tests.testCalculatorCanSumSquareTriangleAndCircleAreasCorrectly();
   return EXIT_SUCCESS;
}
```