```
#include <iostream>
#include <cmath>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include "sierpinski.hpp"

void Sierpinski::draw(sf::RenderTarget& target, sf::RenderStates states) const{
        target.draw(triangle, states);
        sierpinski(triangle, depth_, target);

}

Sierpinski::Sierpinski(int N, int size_tri){

        side_ = size_tri;
        depth_ = N;


        triangle.setPointCount(3);
        triangle.setPoint(0, sf::Vector2f(0,side_ *(sqrt(3)/2)));//left
        triangle.setPoint(1, sf::Vector2f(side_,side_*(sqrt(3)/2)));//right
        triangle.setPoint(2, sf::Vector2f((side_/2),0));//top


        triangle.setFillColor(sf::Color::Red);
        triangle.setOutlineColor(sf::Color::Green);
        triangle.setOutlineThickness(5);
}


void Sierpinski::sierpinski(sf::ConvexShape mid_triangle, int recursion,sf::RenderTarget& target) const{
        sf::Vector2f left,right,top;
        sf::Vector2f mid_lefttop, mid_leftright, mid_topright;
        sf::ConvexShape temp1_tri, temp2_tri, temp3_tri;

        if(recursion == 0){
                return;
        }
        else{
                left = mid_triangle.getPoint(0);
                right = mid_triangle.getPoint(1);
                top = mid_triangle.getPoint(2);

                mid_lefttop.x = (left.x + top.x)/2;
                mid_lefttop.y = (left.y + top.y)/2;

                mid_leftright.x = (left.x + right.x)/2;
                mid_leftright.y = (left.y + right.y)/2;

                mid_topright.x = (top.x + right.x)/2;
                mid_topright.y = (top.y + right.y)/2;

                temp1_tri = filledtriangle(mid_lefttop,mid_leftright, mid_topright,target);
                temp2_tri = temp1_tri;
                temp3_tri = temp1_tri;

                temp1_tri.setPoint(2,left);
                temp2_tri.setPoint(0,right);
                temp3_tri.setPoint(1,top);

                sierpinski(temp1_tri, recursion - 1, target);
                sierpinski(temp2_tri, recursion - 1, target);
                sierpinski(temp3_tri, recursion - 1, target);




        }



}

sf::ConvexShape Sierpinski::filledtriangle(sf::Vector2f left_tri, sf::Vector2f bottom_tri, sf::Vector2f right_tri,sf::RenderTarget& target) const{

        sf::ConvexShape small_triangle;

        small_triangle.setPointCount(3);
        small_triangle.setPoint(0, left_tri);
        small_triangle.setPoint(1, bottom_tri);
        small_triangle.setPoint(2, right_tri);

        small_triangle.setFillColor(sf::Color::Black);

        target.draw(small_triangle);

        return small_triangle;
}

Sierpinski::˜Sierpinski(){

}
```