```
//
//  sierpinski.cpp
//  ps1
//
//  Created by Jingxian Shi on 1/30/18.
//  Copyright Â© 2018 Jingxian Shi. All rights reserved.
//

#include <iostream>
#include <cmath>
#include "sierpinski.hpp"

Sierpinski::Sierpinski(int depth, double side)
{
    recursion_depth = depth;
    double height = (double)((sqrt(3.)/2.0)*(double)side);
    _top = sf::Vector2f(side/2, 0);
    _left = sf::Vector2f(0, height);
    _right = sf::Vector2f(side, height);

    sf::Vector2f top_left = sf::Vector2f((_top.x+_left.x)/2, (_left.y+_top.y)/2);
    sf::Vector2f top_right = sf::Vector2f((_right.x+_top.x)/2, (_right.y+_top.y)/2);
    sf::Vector2f bottom_mid = sf::Vector2f((_right.x+_left.x)/2, (_right.y+_left.y)/2);

    setPointCount(3);
    setPoint(0, top_left);
    setPoint(1, top_right);
    setPoint(2, bottom_mid);
    setFillColor(sf::Color::Black);
}

Sierpinski::Sierpinski(int depth, sf::Vector2f top, sf::Vector2f left, sf::Vector2f right)
{
    recursion_depth = depth;
    _top = top;
    _left = left;
    _right = right;

    sf::Vector2f top_left = sf::Vector2f((_top.x+_left.x)/2, (_left.y+_top.y)/2);
    sf::Vector2f top_right = sf::Vector2f((_right.x+_top.x)/2, (_right.y+_top.y)/2);
    sf::Vector2f bottom_mid = sf::Vector2f((_right.x+_left.x)/2, (_right.y+_left.y)/2);

    setPointCount(3);
    setPoint(0, top_left);
    setPoint(1, top_right);
    setPoint(2, bottom_mid);
    setFillColor(sf::Color::Black);
}

void Sierpinski::draw(sf::RenderTarget& target, sf::RenderStates states) const
{
    target.draw((sf::ConvexShape)(*this), states);
    if(recursion_depth <= 0)
    {
        return;
    }
    else
    {
        Sierpinski top(recursion_depth-1,
                       _top, //top
                       sf::Vector2f((_top.x+_left.x)/2.0, (_left.y+_top.y)/2.0), //bottom left
                       sf::Vector2f((_right.x+_top.x)/2.0, (_right.y+_top.y)/2.0)); //bottom right
        top.draw(target, states);

        Sierpinski left(recursion_depth-1,
                        sf::Vector2f((_top.x+_left.x)/2.0, (_top.y+_left.y)/2.0),
                        _left,
                        sf::Vector2f((_left.x+_right.x)/2.0, (_left.y+_right.y)/2.0));
        left.draw(target, states);

        Sierpinski right(recursion_depth-1,
                         sf::Vector2f((_right.x+_top.x)/2, (_right.y+_top.y)/2),
                         sf::Vector2f((_right.x+_left.x)/2, (_right.y+_left.y)/2),
                         _right);
        right.draw(target, states);
    }
}


```

```
//
//  sierpinski.hpp
//  ps1
//
//  Created by Jingxian Shi on 1/30/18.
//  Copyright Â© 2018 Jingxian Shi. All rights reserved.
//

#ifndef sierpinski_hpp
#define sierpinski_hpp

#include <stdio.h>
#include <SFML/Graphics.hpp>
#include <SFML/Graphics/ConvexShape.hpp>
#include <iostream>

using namespace std;

class Sierpinski : public sf::ConvexShape
{
public:
    Sierpinski(int depth, double side);
    Sierpinski(int depth, sf::Vector2f top, sf::Vector2f left, sf::Vector2f right
);
    virtual void draw(sf::RenderTarget& target, sf::RenderStates states) const;
private:
    int recursion_depth;
    sf::Vector2f _top, _left, _right;
};

#endif /* sierpinski_hpp */


```