

# 计算机学院\_高级语言程序设计\_课程实验报告

实验题目：空间数据		学号：202200130048
日期：2023. 5. 26	班级： 6	姓名： 陈静雯
Email：1205037094@qq. com		
<p>实验步骤与内容：</p> <pre>#include &lt;iostream&gt; #include &lt;vector&gt; #include &lt;cstring&gt; #include &lt;cmath&gt; using namespace std;  template&lt;typename T&gt; class Point{ private:     T x;     T y;     T z; public:     friend class Polyhedron;     friend class Facet;     Point(T x, T y, T z) : x(x), y(y), z(z) {}     void copy(const Point&lt;T&gt;&amp; p) {         x=p. x;         y=p. y;         z=p. z;     }     T distant(Point&lt;T&gt; a) {         return sqrt(pow(x-a. x, 2)+pow(y-a. y, 2)+pow(z-a. z, 2));     }     ~Point() {} };  template&lt;typename T&gt; class Facet{ private:     vector&lt;Point&lt;T&gt;*&gt;vertices; public:     Facet() {}     void get(Point&lt;T&gt;* x) {         vertices. push_back(x);     } };</pre>		

```

    }
    void clear() {
        vertices.clear();
    }
    Point<T> get_low() {
        for(auto i =vertices.begin();i!=vertices.end();++i) {
            Point<T> min(0.0,100000.0,0.0);
            if(i.y<min.y) min.copy(i);
        }
    }
    Point<T> get_high() {
        for(auto i =vertices.begin();i!=vertices.end();++i) {
            Point<T> max(0.0,-100000.0,0.0,0);
            if(i.y>max.y) max.copy(i);
        }
    }
    T area() {
        for(auto i =vertices.begin()+1;i!=vertices.end()-1;++i) {
            T a = vertices[0]->distant(*i);
            T b = vertices[0]->distant(*i);
            T c = i->distant(i+1);
            T p = (a+b+c)/2;
            return sqrt(p*(p-a)*(p-b)*(p-c));
        }
    }
    friend class Polyhedron;
    ~Facet() {
        for(auto i =vertices.begin();i!=vertices.end();++i)
            delete *i;
    }
};

```

```

template<typename T>
class Polyhedron{
private:
    vector<Point<T>*>vertices;
    vector<Facet<T>*>facets;
public:
    PolyhedronFILE (const char *path) {
        FILE *fp(path);
        string str;
        fscanf(fp,"%[^\\n]",str);
        if(str=="OFF") {
            int a,b,c;

```

```

        fscanf(fp, "%d", &a);
        fscanf(fp, "%d", &b);
        fscanf(fp, "%d", &c);
        int i=1;
        float p[10];
        while(i<=a) {
            for(int j=0;j<3;j++) fscanf(fp, "%f", &p[j]);
            vertices.push_back(new Point<T>(p[0], p[1], p[2]));
        }
        i=1;
        while(i<=b) {
            Facet<T>* fac = new Facet<T>;
            int l;
            fscanf(fp, "%d", &l);
            int f[l];
            for(int j=0;j<l;j++) {
                fscanf(fp, "%d", &f[j]);
                fac->get(vertices[f[j]]);
            }
            facets.push_back(fac);
        }
    }
    else{
        cout<<" illeagle file "<<endl;
    }
}

T area() {
    T s = 0;
    for(auto i=facets.begin(); i!=facets.end(); ++i) {
        s+=i->area();
    }
    return s;
}

~Polyhedron() {
    for(auto i =vertices.begin(); i!=vertices.end(); ++i) {
        delete *i;
    }
    for(auto i =facets.begin(); i!=facets.end(); ++i) {
        delete *i;
    }
}

};

```

```
int main() {  
    char path[100]="3holes.off";  
    Polyhedron<float> pol(path); //不知道传这个 const char*的参数该怎么传  
    float s = pol.area();  
    cout<<s;  
}
```

结论分析与体会：

先读取一行看是否是 OFF，再根据第二行的点面个数读取后面的一个个数据，最后利用海伦公式先算一个面的面积再相加。