图形结构

#了解自定义类型的概念和用途

设计一个图形结构体，其包含一个 enum 枚举类型，表示图形的种类，其值可以为 `Circle`、`Rectangle` 和 `Triangle`，然后还包含一个 union 共用体，存储不同种类的对应参数，例如圆形的半径，矩形的宽度和高度、三角形的底边和高等。实现一个函数，用来计算这些图形的面积

依次输入圆、矩形和三角形的相应参数，输出三个图形的面积，如果数据非法，输出 `error`

输入样例 1

```
5 2 3 4 6
```

输出样例 1

```
Circle area: 78.5398
Rectangle area: 6
Triangle area: 12
```

输入样例 2

```
-1 1 2 3 4
```

输出样例 2

```
Circle area: error
Rectangle area: 2
Triangle area: 6
```

样例

```
0 0 0 0 0
```

```
Circle area: 0
Rectangle area: 0
Triangle area: 0
```

```
-1 -1 -1 -1 -1
```

```
Circle area: error
Rectangle area: error
Triangle area: error
```

```
56 71 25 36 78
```

```
Circle area: 9852.03
Rectangle area: 1775
Triangle area: 1404
```

```
3.7 4.6 3.8 1.4 2.9
```

```
Circle area: 43.0084
Rectangle area: 17.48
Triangle area: 2.03
```

Codes

```cpp
#include <iostream>
#include <cmath> // 用于 sqrt 和 pow 函数

// 枚举类型，定义图形种类
enum ShapeType {
    Circle,
    Rectangle,
    Triangle
};

// 共用体，用于存储不同类型图形的数据
union ShapeData {
    struct {
        double radius; // 圆形的半径
    } circle;
    struct {
        double width, height; // 矩形的宽度和高度
    } rectangle;
    struct {
        double base, height; // 三角形的底边和高
    } triangle;
};

// 结构体，表示一个图形
struct Shape {
    ShapeType type;
```

```
    ShapeData data;

    // 计算面积的成员函数
    double area() const {
        switch (type) {
            case Circle:
                if (data.circle.radius < 0) return -1;
                return M_PI * data.circle.radius * data.circle.radius;
            case Rectangle:
                if (data.rectangle.width < 0 || data.rectangle.height < 0) return
-1;
                return data.rectangle.width * data.rectangle.height;
            case Triangle:
                if (data.triangle.base < 0 || data.triangle.height < 0) return -1;
                return 0.5 * data.triangle.base * data.triangle.height;
            default:
                return 0.0; // 默认情况处理
        }
    }
};

int main() {
    double a, b, c, d, e;
    std::cin >> a >> b >> c >> d >> e;
    // 创建图形实例
    Shape circle = {Circle, { .circle = {a} }};
    Shape rectangle = {Rectangle, { .rectangle = {b, c} }};
    Shape triangle = {Triangle, { .triangle = {d, e} }};

    // 输出每个图形的面积
    if (circle.area() == -1)
        std::cout << "Circle area: error"  << std::endl;
    else
        std::cout << "Circle area: " << circle.area() << std::endl;

    if (rectangle.area() == -1)
        std::cout << "Rectangle area: error"  << std::endl;
    else
        std::cout << "Rectangle area: " << rectangle.area() << std::endl;


    if (triangle.area() == -1)
        std::cout << "Triangle area: error"  << std::endl;
    else
        std::cout << "Triangle area: " << triangle.area() << std::endl;

    return 0;
}
```