개요 | 제출 | 편집 | 코딩 결과

실습 102 - Shape Hierarchy

제출 마감일: 2023-06-09 23:59
업로드 가능한 파일 수: 4
제출 방식: 개인

Problem

Consider the provided IShape interface and its implementations. Your task is to create a Shape hierarchy by designing and implementing three specific shapes, Rectangle, Triangle, and now Circle. Each of these shapes should inherit from IShape.

- The Rectangle class should ensure that it takes four points to construct a rectangle. Implement the getArea(), getPerimeter(), and toString() functions specific to a rectangle.
- The Triangle class should ensure that it takes three points to construct a triangle. Implement the getArea(), getPerimeter(), and toString() functions specific to a triangle.
- The Circle class should ensure that it takes a center point and a radius to construct a circle. Implement the getArea(), getPerimeter(), and toString() functions specific to a circle.

<참고자료>

```
// IShape.h

class Point {
public:
int x = 0;
int y = 0;
Point(int _x, int _y) : x(_x), y(_y) {};
};

class IShape {
public:
IShape() = default;
virtual ~IShape() noexcept {
std::cout << "IShape destructor called" << std::endl;
};
IShape(const std::vector<Point>& points) : points(points) {};;
virtual double getArea() const = 0;
virtual double getPerimeter() const = 0;

friend std::ostream& operator<<(std::ostream& str, const IShape& shape);
```

```
protected:
virtual std::string toString() const = 0;
std::vector<Point> points;
};

std::ostream& operator << (std::ostream& os, const IShape& shape) {
return os << shape.toString();
}


//ShapeTest.cpp

void doingSomething(const vector<unique_ptr<IShape>>& shapes) {
for (const auto& s : shapes)
cout << *s << endl;

double totalArea = accumulate(shapes.cbegin(), shapes.cend(), 0.0,
[](double cur_sum, const auto& rhs){
return cur_sum + (*rhs).getArea();
});

double totalPerimeter = accumulate(shapes.cbegin(), shapes.cend(), 0.0,
[](double cur_sum, const auto& rhs){
return cur_sum + (*rhs).getPerimeter();
});
cout << std::fixed << std::setprecision(2);
cout << "totalArea: " << totalArea << ", " << "totalPerimeter: " << totalPerimeter << endl;
}


int main() {
std::vector<std::unique_ptr<IShape>> shapes;

shapes.emplace_back(std::make_unique<Rectangle>(std::vector<Point>{{0, 0}, {0, 2}, {2, 2}, {2, 0}}));
shapes.emplace_back(std::make_unique<Triangle>(std::vector<Point>{{0, 0}, {0, 2}, {2, 2}}));
shapes.emplace_back(std::make_unique<Circle>(Point{0, 0}, 3));

doingSomething(shapes);

return 0;
}

//Helper.h

double calcDist(const Point& p1, const Point& p2) {
int dx = p2.x - p1.x;
int dy = p2.y - p1.y;
return std::sqrt(dx*dx + dy*dy);
}
```

입력

없음

출력

Rectangle Area: 4.00, Perimeter: 8.00

Triangle Area: 2.00, Perimeter: 6.83

Circle Area: 28.27, Perimeter: 18.85

totalArea: 34.27, totalPerimeter: 33.68

Rectangle destructor called

IShape destructor called

Triangle destructor called

IShape destructor called

Circle destructor called

IShape destructor called

제출파일

Circle.h
Rectangle.h
Triangle.h
102.csv

(IShape.h, Helper.h, ShapeTest.cpp 파일은 PLATO 서버에 등록되어 있습니다)

VPL