**HOMEWORK** : 13/1/2025

**NAME** : Võ Thị Thúy Sang - 22IT248

**CLASS** : Lập trình web nâng cao (1)

**Bài thực hành số 3:**

#include <iostream>

#include <vector>

#include <sstream>

#include <locale>

using namespace std;

class Shape {

protected:

int x, y;

public:

Shape(int x, int y) : x(x), y(y) {}

virtual void Move(int newX, int newY) {

x = newX;

y = newY;

}

virtual string ToString() const = 0;

virtual void Show() const = 0;

virtual ~Shape() {}

};

class Line : public Shape {

private:

int x2, y2;

public:

Line(int x1, int y1, int x2, int y2) : Shape(x1, y1), x2(x2), y2(y2) {}

string ToString() const override {

stringstream ss;

ss << "Line: (" << x << ", " << y << ") to (" << x2 << ", " << y2 << ")";

return ss.str();

}

void Show() const override {

cout << ToString() << endl;

}

};

class Circle : public Shape {

private:

int radius;

public:

Circle(int x, int y, int r) : Shape(x, y), radius(r) {}

string ToString() const override {

stringstream ss;

ss << "Circle: Center(" << x << ", " << y << "), Radius: " << radius;

return ss.str();

}

void Show() const override {

cout << ToString() << endl;

}

};

class Rectangle : public Shape {

private:

int width, height;

public:

Rectangle(int x, int y, int w, int h) : Shape(x, y), width(w), height(h) {}

string ToString() const override {

stringstream ss;

ss << "Rectangle: Top-left(" << x << ", " << y << "), Width: " << width << ", Height: " << height;

return ss.str();

}

void Show() const override {

cout << ToString() << endl;

}

};

class PolyLine : public Shape {

private:

vector<pair<int, int>> points;

public:

PolyLine(int x, int y) : Shape(x, y) {

points.push\_back({ x, y });

}

void AddPoint(int x, int y) {

points.push\_back({ x, y });

}

string ToString() const override {

stringstream ss;

ss << "PolyLine: ";

for (const auto& p : points) {

ss << "(" << p.first << ", " << p.second << ") ";

}

return ss.str();

}

void Show() const override {

cout << ToString() << endl;

}

};

int main() {

Line line(0, 0, 10, 10);

line.Show();

Circle circle(5, 5, 3);

circle.Show();

Rectangle rect(2, 2, 6, 4);

rect.Show();

PolyLine poly(1, 1);

poly.AddPoint(3, 4);

poly.AddPoint(5, 6);

poly.Show();

return 0;

}