

Лабораторна робота

Розробка програм з користувальськими класами. Робота з класами та об'єктами.

Виконав: Вівчар Вадим Вікторович, група ALK-43

Мета роботи: Навчитися працювати з класами та об'єктами. Набути навичок об'єктно-орієнтованого програмування.

Завдання: Створити абстрактний клас Figure для обчислення об'єму та площи поверхні геометричних тіл. Реалізувати похідні класи Cone, Cylinder, Sphere з відповідними методами.

Код програми:

```
#include <iostream>
#include <cmath>
using namespace std;

class Figure {
protected:
    double x, y;
public:
    Figure(double x = 0, double y = 0) : x(x), y(y) {}
    virtual double volume() const = 0;
    virtual double area() const = 0;
    virtual void print() const = 0;
    virtual ~Figure() {}
};

class Cone : public Figure {
    double radius, height;
public:
    Cone(double r, double h) : radius(r), height(h) {}
    double volume() const override { return M_PI * radius * radius * height / 3.0; }
    double area() const override { return M_PI * radius * (radius + height); }
    void print() const override {
        cout << "Cone: r=" << radius << ", h=" << height << endl;
    }
};

class Cylinder : public Figure {
    double radius, height;
public:
    Cylinder(double r, double h) : radius(r), height(h) {}
    double volume() const override { return M_PI * radius * radius * height; }
    double area() const override { return 2 * M_PI * radius * (radius + height); }
    void print() const override {
        cout << "Cylinder: r=" << radius << ", h=" << height << endl;
    }
};
```

```

class Sphere : public Figure {
    double radius;
public:
    Sphere(double r) : radius(r) {}
    double volume() const override { return 4.0 / 3.0 * M_PI * pow(radius, 3); }
    double area() const override { return 4 * M_PI * radius * radius; }
    void print() const override {
        cout << "Sphere: r=" << radius << endl;
    }
};

int main() {
    Cone cone(3, 5);
    Cylinder cylinder(3, 5);
    Sphere sphere(3);

    cone.print();
    cout << "Volume: " << cone.volume() << ", Area: " << cone.area() << endl;

    cylinder.print();
    cout << "Volume: " << cylinder.volume() << ", Area: " << cylinder.area() << endl;

    sphere.print();
    cout << "Volume: " << sphere.volume() << ", Area: " << sphere.area() << endl;

    return 0;
}

```

Результат виконання програми:

На скріншоті показано результати виконання програми для трьох реалізованих класів: Cone, Cylinder і Sphere.

```

D:\work\Project5.exe
Cone: r=3, h=5
Volume: 47.1239, Area: 83.2298

Cylinder: r=3, h=5
Volume: 141.372, Area: 150.796

Sphere: r=3
Volume: 113.097, Area: 113.097

-----
Process exited after 0.1051 seconds with return value 0
Press any key to continue . . .

```

Контрольні запитання:

1. Принцип ООП — успадкування: створення класу на основі

іншого.

2. Специфікатори доступу: public, protected, private.
3. При public-успадкуванні відкриті залишаються відкритими.
4. При private-успадкуванні всі члени стають приватними.
5. Protected дозволяє доступ похідним класам, але не зовнішнім функціям.
6. Конструктори викликаються від базового до похідного, деструктори — навпаки.
7. Множинне успадкування — наслідування від кількох базових класів.