Молдавский Государственный Университет Молдовы
Факультет Математики и Информатики
Департамент Информатики

Лабораторная работа №2

по предмету “Объектно-ориентированное программирование”
тема:”Наследование”

Преподаватель: Латул Г.
Выполнила: Павлышина Александра I2302

Кишинев, 2024

Вариант 8. Создайте иерархию классов Сторона – Квадрат – Пирамида. Класс Пирамида должен содержать метод для определения объема пирамиды. Последние два класса в иерархии должны иметь конструкторы. Создайте метод MAIN, в котором создается 2 пирамиды, определяется какая из пирамид большая, а также входит ли меньшая пирамида полностью в большую. Необходимо также показывать все характеристики создаваемых объектов.

Листинг программы.

Класс Side (сторона) :

```
class Side {
protected:
    double length;

public:
    Side(double len) : length(len) {} // конструктор, присваивающий значение длины стороны свойству length

    double getLength() { // метод, возвращающий длину стороны
        return length;
    }
};
```

Класс Square (квадрат), наследуемый от класса Side :

```
class Square : public Side {
public:
    Square(double len) : Side(len) {} /* конструктор, который вызывает конструктор базового класса Side
    и передает ему длину стороны.*/

    double area() { /* метод который возвращает площадь квадрата, вычисляемую как произведение длины
    стороны на саму себя.*/
        return length * length;
    }
};
```

Класс Pyramid (пирамида), наследуемый от класса Square :

```
class Pyramid : public Square {
private:
    double height;

public:
    Pyramid(double len, double h) : Square(len), height(h) {} /* конструктор, который вызывает конструктор
    класса Square и инициализирует как длину стороны, так и высоту пирамиды.*/

    double getHeight() {
        return height;
    }

    double volume() {
        return (1.0 / 3.0) * Square::area() * height; /* объем пирамиды равен трети произведения площади
        ее основания и высоты.*/
    }
};
```

Метод main(), в котором содержатся данные о пирамидах и сравнение их объема :

```
int main() {
    Pyramid pyramid1(10.0, 5.0); // (длина, высота)
    Pyramid pyramid2(5.0, 10.0);

    cout << "Информация о пирамиде 1:" << endl;
    cout << "Высота: " << pyramid1.getHeight() << endl;
    cout << "Длина стороны: " << pyramid1.getLength() << endl;
    cout << "Объем: " << pyramid1.volume() << endl;

    cout << "\nИнформация о пирамиде 2:" << endl;
    cout << "Высота: " << pyramid2.getHeight() << endl;
    cout << "Длина стороны: " << pyramid2.getLength() << endl;
    cout << "Объем: " << pyramid2.volume() << endl;

    if (pyramid1.volume() > pyramid2.volume()) {
        cout << "\nПирамида 1 больше." << endl;
        cout << "Пирамида 2 может поместиться в пирамиду 1." << endl;
    } else if (pyramid1.volume() < pyramid2.volume()) {
        cout << "\nПирамида 2 больше." << endl;
        cout << "Пирамида 1 может поместиться в пирамиду 2." << endl;
    } else {
        cout << "\nОбе пирамиды имеют одинаковый объем." << endl;
    }

    return 0;
}
```

Результат :

```
Информация о пирамиде 1:
Высота: 5
Длина стороны: 10
Объем: 166.667

Информация о ?ирамиде 2:
Высота: 10
Длина стороны: 5
Объем: 83.3333

Пи?амида 1 больше.
Пирамида 2 может поместиться в пирам?ду 1.


...Program finished with exit code 0
Press ENTER to exit console.
```

Используемые библиотеки.

#include <iostream> - библиотека для работы с вводом и выводом (cin, cout).

Используемые функции.

Side(double len) : length(len) {} - конструктор, присваивающий значение длины стороны свойству length

Square(double len) : Side(len) {} - конструктор, который вызывает конструктор базового класса Side и передает ему длину стороны.

Pyramid(double len, double h) : Square(len), height(h) {} - конструктор, который вызывает конструктор класса Square и инициализирует как длину стороны, так и высоту пирамиды.