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) {}

    double getLength() const {
        return length;
    }
};
```

Класс Square (квадрат), наследуемый от класса Side :

```
class Square : public Side {
public:
    Square(double len) : Side(len) {}

    double area() const {
        return length * length;
    }
};
```

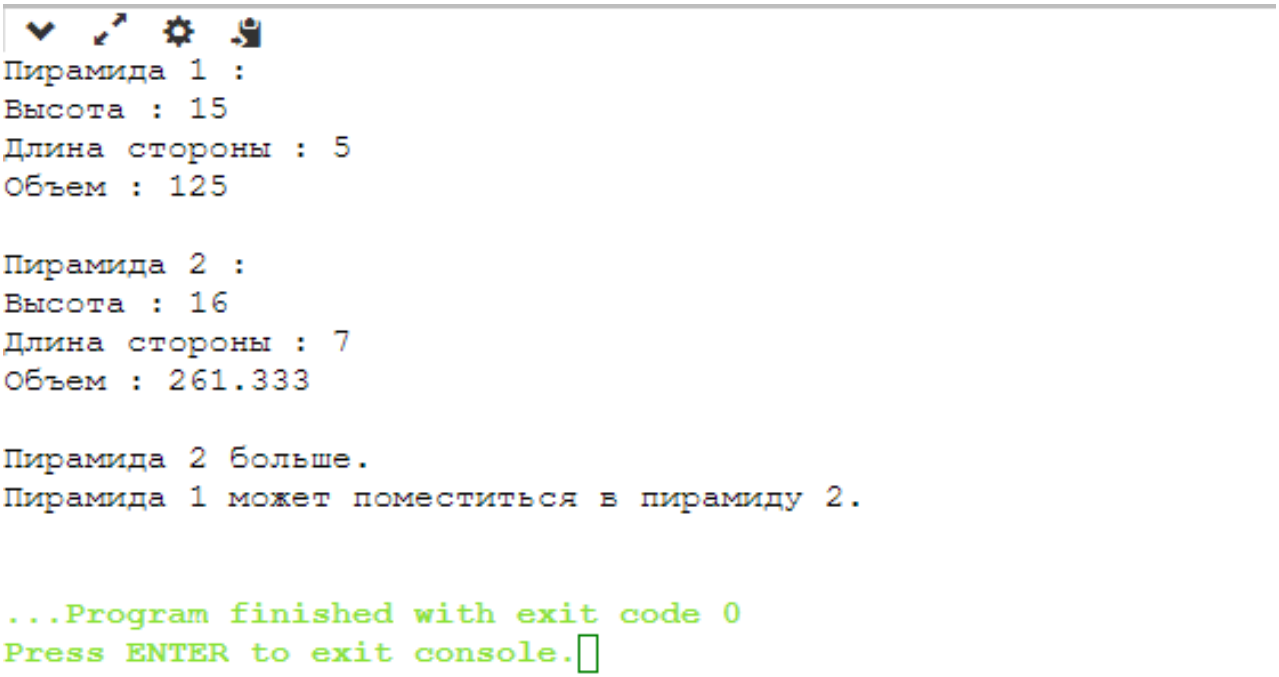
Класс Pyramid (пирамида), наследуемый от класса Square :

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

public:
    Pyramid(double length, double h) : Square(length), height(h) {}

    double getHeight() const {
        return height;
    }

    double volume() const {
        return (1.0 / 3.0) * Square::area() * height;
    }

    bool canContain(const Pyramid& other) const {
        return (area() >= other.area() && getHeight() >= other.getHeight());
    }
};
```

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

```cpp
int main() {
    Pyramid pyramid1(5.0, 15.0);
    Pyramid pyramid2(7.0, 16.0);

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

    cout << "\nПирамида 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;
        if (pyramid1.canContain(pyramid2)) // если true
            cout << "Пирамида 2 может поместиться в пирамиду 1." << endl;
        else
            cout << "Пирамида 2 не может поместиться в пирамиду 1." << endl;
    } else if (pyramid1.volume() < pyramid2.volume()) {
        cout << "\nПирамида 2 больше." << endl;
        if (pyramid2.canContain(pyramid1))
            cout << "Пирамида 1 может поместиться в пирамиду 2." << endl;
        else
            cout << "Пирамида 1 не может поместиться в пирамиду 2." << endl;
    } else {
        cout << "\nУ обеих пирамид одинаковый объем." << endl;
    }

    return 0;
}
```

Результат :



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.

bool canContain(const Pyramid& other) - булевая функция, которая сравнивает площади двух пирамид