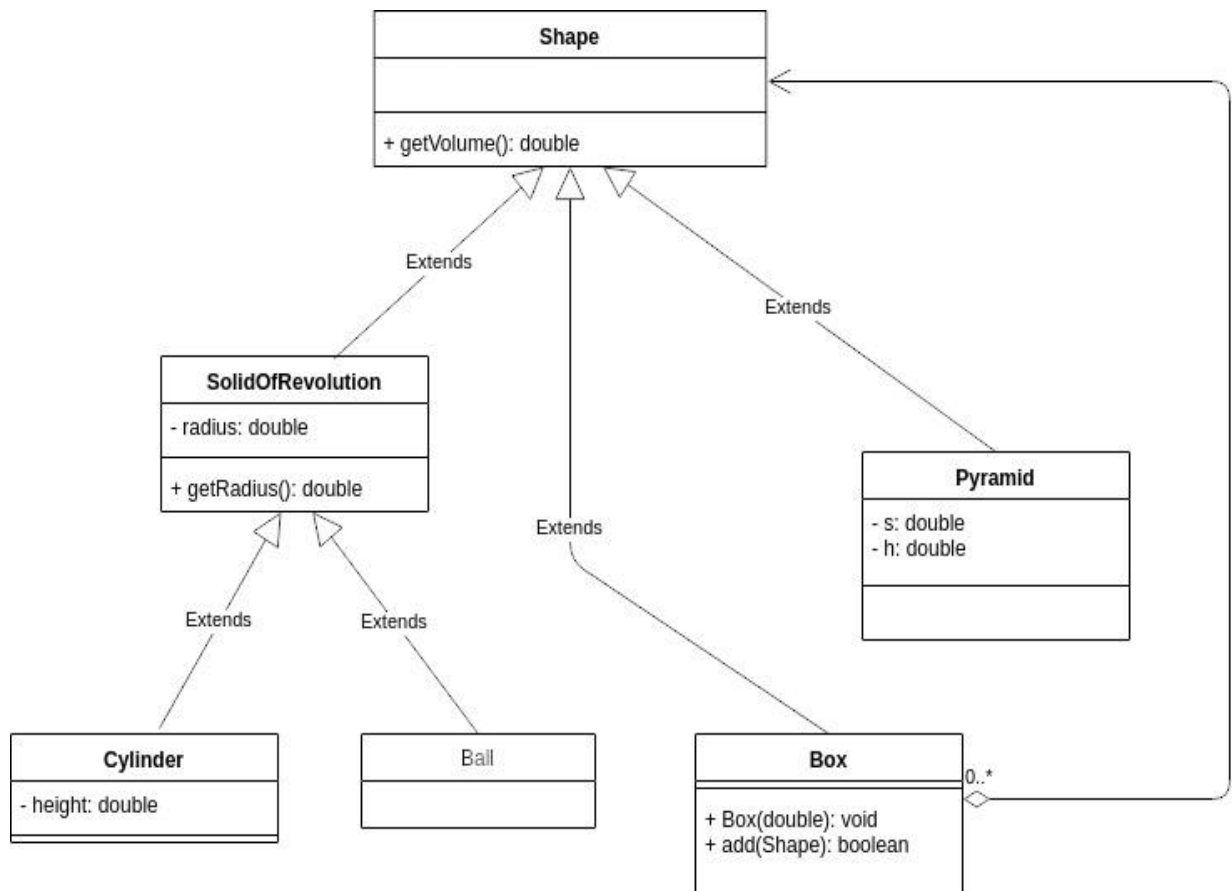


## Задание 4

### Вариант 1

Реализовать иерархию классов, описывающую трёхмерные фигуры.



Классы Shape и SolidOfRevolution реализовать как абстрактные.

Класс Box – это «контейнер», он «содержит» в себе другие фигуры. Метод add() принимает на вход Shape. Нужно добавлять новые фигуры до тех пор, пока для них хватает места в Box (будем считать только объём, игнорируя форму. Допустим, мы переливаем жидкость). Добавленные фигуры хранятся в классе Box в виде массива или поля типа ArrayList<Shape>. Если места для добавления новой фигуры не хватает, то метод должен вернуть false.

Для удобства вывода информации об объектах рекомендуется определить в классах метод toString().

### Вариант 2

Пусть имеется следующий класс MyStack. Нужно добавить в него метод сортировки объектов. Метод сортировки сортирует объекты используя метод

`compareTo()`. УКАЗАНИЕ: Следует изменить тип объектов в стеке с `Object` на `Comparable`.

Протестировать стек на данных класса, реализующего интерфейс `Comparable` и свой метод `compareTo()`, например, на классах-обёртках чисел или на классе `String`.

```
import java.util.ArrayList;

public class MyStack {
    private ArrayList<Object> list = new ArrayList<>();

    public boolean isEmpty() {
        return list.isEmpty();
    }

    public int getSize() {
        return list.size();
    }

    public Object peek() {
        return list.get(getSize() - 1);
    }

    public Object pop() {
        Object o = list.get(getSize() - 1);
        list.remove(getSize() - 1);
        return o;
    }

    public void push(Object o) {
        list.add(o);
    }

    @Override /** Переопределяет метод toString класса Object */
    public String toString() {
```

```
        return "чек: " + list.toString();  
    }  
}
```