Полиморфизм

**Проблема:**

Зачастую возникают ситуации, когда необходимо описать несколько объектов, которые имеют общие черты, но отличаются в поведении.

**Примерами могут быть:**

Животные – все умеют подавать голос, но каждое делает это по-своему:

* **Корова мычит**
* **Собака лает**

Транспортные средства – все умеют перемещаться, но:

* **Корабль плывет**
* **Машина едет**

Рассмотрим задачу:

**Фигуры**

Мы начинаем делать аналог AutoCAD – программа для чертежей.

На самом раннем этапе мы реализуем примитивы: линия, треугольник, прямоугольник, круг.

Нам необходимо иметь возможность посчитать площадь, периметр, а также задавать цвет каждой из фигур.

**Процедурный подход**

Определять тип фигуры и выбирать соответствующую формулу:

double getArea(Figure figure) {  
 if (CIRCLE.equals(figure.getType())){  
 // π\*R\*R  
 } else if (RECTANGLE.equals(figure.getType())) {  
 //a\*b  
 }else if ...  
}

**ООП-подход**

**Для этого предлагаю создать иерархию классов:**

* Базовый класс Fugure
* Line
* Triange
* Rectangle
* Circle

**Заглушка**

Color – класс из фреймворка Android. На данный момент создадим для него заглушку, хранящую только имя цвета

class Color {  
 private String name;  
 public Color (String name){  
 this.name = name;  
 }  
 public String getName() {return name;}  
}

**Периметр и площадь**

Периметр и площадь мы можем рассчитать для каждой из фигур.

**Проблема:**

* Функции должны быть объявлены в базовом классе, так как присущи всем объектам
* Для разных объектов функции должны работать по-разному

Понятие полиморфизма

Дословный перевод – много форм

Это способность объекта выглядеть, как его предок, но отличаться в поведении