

# Паттерн Composite

Есилевич Александр

18 ноября 2011 г.

# Перемещение фигур

```
public class Shape {  
    public abstract void move(int dx, int dy);  
}
```

# Перемещение фигур

```
public class Line extends Shape {  
    public void move(int dx, int dy) {  
        x1 = x1 + dx;  
        x2 = x2 + dx;  
        y1 = y1 + dy;  
        y2 = y2 + dy;  
    }  
}
```

# Перемещение фигур

```
public class Rectangle extends Shape {  
    public void move(int dx, int dy) {  
        x1 = x1 + dx;  
        x2 = x2 + dx;  
        y1 = y1 + dy;  
        y2 = y2 + dy;  
    }  
}
```

## Перемещение фигур

```
public class Circle extends Shape {  
    public void move(int dx, int dy) {  
        x = x + dx;  
        y = y + dy;  
    }  
}
```

## Группировка фигур

Добавим класс ShapeGroup, который будет представлять группу фигур.

```
public class ShapeGroup extends Shape {  
    public void addShape(Shape s);  
    public void removeShape(Shape s);  
  
    public void draw(SystemCanvas canvas);  
    public void move(int dx, int dy);  
  
    private ShapeList shapes;  
}
```

## Отрисовка группы фигур

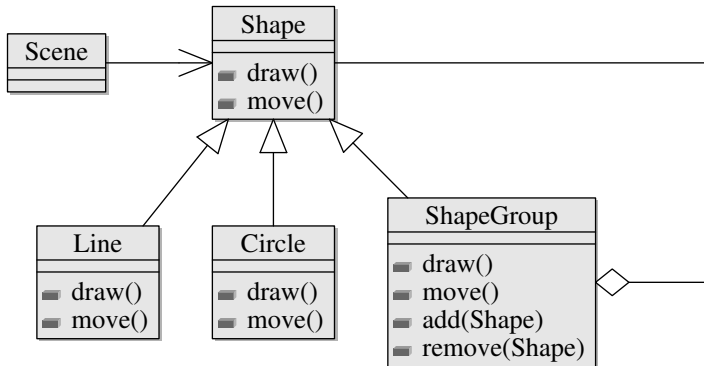
```
public class ShapeGroup extends Shape {  
  
    public void draw(SystemCanvas canvas) {  
        ShapeListIterator it = shapes.getFirst();  
        while(!it.isEnd()) {  
            it.getShape().draw(graphics);  
            it = it.getNext();  
        }  
    }  
}
```

## Перемещение группы фигур

```
public class ShapeGroup extends Shape {  
  
    public void move(int dx, int dy) {  
        ShapeListIterator it = shapes.getFirst();  
        while(!it.isEnd()) {  
            it.getShape().move(dx, dy);  
            it = it.getNext();  
        }  
    }  
  
}
```



## Диаграмма классов



# Паттерн Composite

Назначение: компонует объекты в древовидные структуры.  
Позволяет клиентам единообразно трактовать индивидуальные и составные объекты.

# Паттерн Composite. Диаграмма классов

