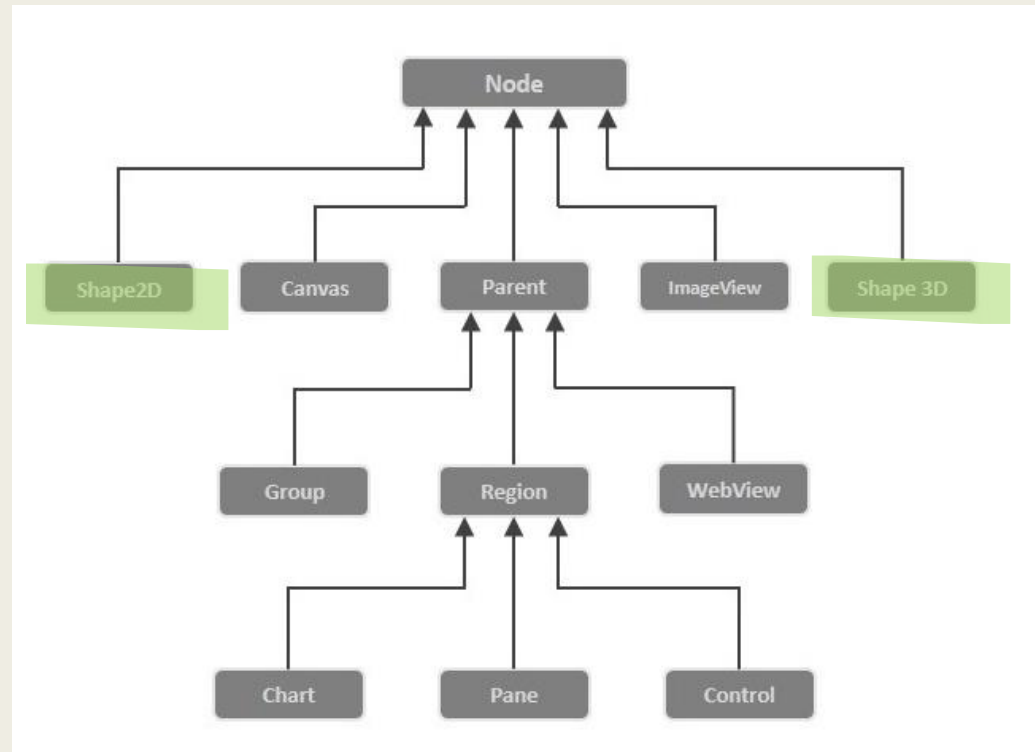


2D ГРАФИКА



javafx.scene.shape.Shape

javafx.scene.shape.Shape3D



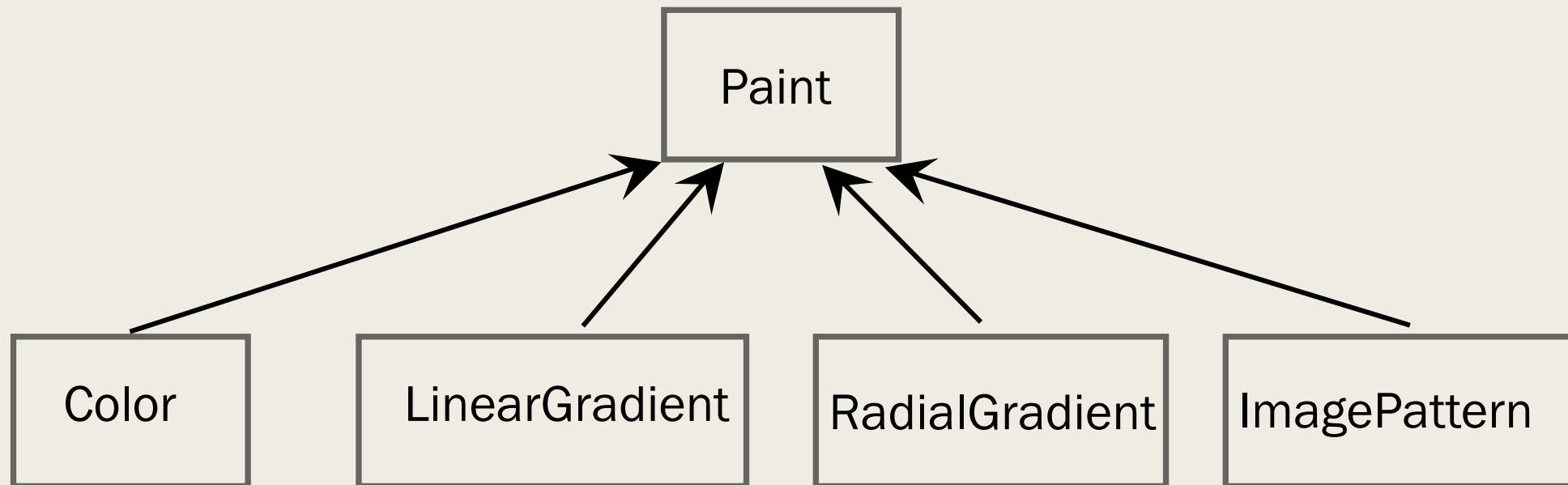
Вспомогательные классы

Модуль `javafx.graphics`

```
import javafx.geometry.*;
```

- `Point2D` – координаты точки
- `Dimension2D` - размеры прямоугольной области
- `Rectangle2D` – координаты и размер прямоугольной области
- `Bounds` – координаты и размер области, в которой расположен узел (абстрактный)
- `BoundingBox` – реализация класса `Bounds`

Свойства фона



Метод класса Scene

setFill(Paint value)

Режимы наложения

- Порядок добавления узлов в контейнер определяет порядок перекрытия при наложении
- В классе Node методы
 - *setBlendMode (BlendMode v)*
 - *BlendMode getBlendMode()*
 - *ObjectProperty<BlendMode> blendModeProperty()*
- Основные режимы BlendMode.SRC_OVER и BlendMode.SRC_ATOP
- Остальные режимы: цвета, затемнения, осветления, контрастности, сравнения
- Изменить порядок следования toBack(), toFront()

Blend.java - пример

Маски слоя

Видимость слоя можно ограничивать с помощью маски. Часть слоя внутри маски будет видна, а оставшаяся часть – скрыта.

В качестве маски можно использовать узел, например – прямоугольник или компонент `ImageView`

Методы

- `void setClip(Node v)`
- `Node getClip()`
- `ObjectProperty<Node> clipProperty()`

`Clip.java` – пример (нужен файл `mask.png`)

Фигуры

Базовый абстрактный класс Shape

- Line
- Arc – дуга или сектор
- Polyline – ломаная линия
- CubicCurve – кубическая кривая Безье
- QuadCurve – квадратичная кривая
- Rectangle
- Ellipse
- Circle
- Polygon

javafx.scene.shape.Shape

- `javafx.scene.shape.Shape`
 - `javafx.scene.shape.Arc`
 - `javafx.scene.shape.Circle`
 - `javafx.scene.shape.CubicCurve`
 - `javafx.scene.shape.Ellipse`
 - `javafx.scene.shape.Line`
 - `javafx.scene.shape.Path`
 - `javafx.scene.shape.Polygon`
 - `javafx.scene.shape.Polyline`
 - `javafx.scene.shape.QuadCurve`
 - `javafx.scene.shape.Rectangle`
 - `javafx.scene.shape.SVGPath`

Характеристики фигур

- Фон
 - *setFill(Paint value)*
- Обводка
 - *setStroke(Paint value)*
 - *setStrokeWidth(double value)*
 - *setStrokeType(StrokeType value)*
 - ...
- Сглаживание контура
 - *setSmooth(boolean value)*

Другие классы

- Path – траектория. Описывает траекторию, состоящую из отдельных линий

```
VLineTo vline = new VLineTo(); vline.setY(200.0);
```

```
HLineTo hline = new HLineTo(); hline.setX(200.0);
```

```
LineTo line = new LineTo();
```

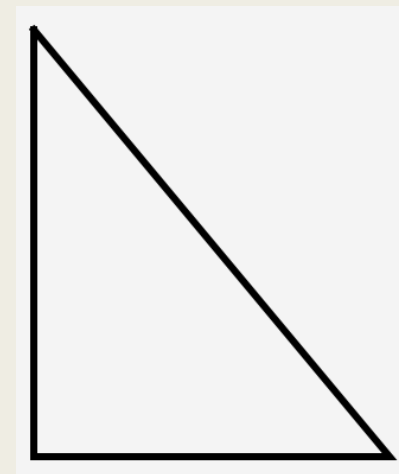
```
line.setX(50.0);
```

```
line.setY(20.0);
```

```
Path p = new Path(new MoveTo(50.0, 20.0), vline, hline, line);
```

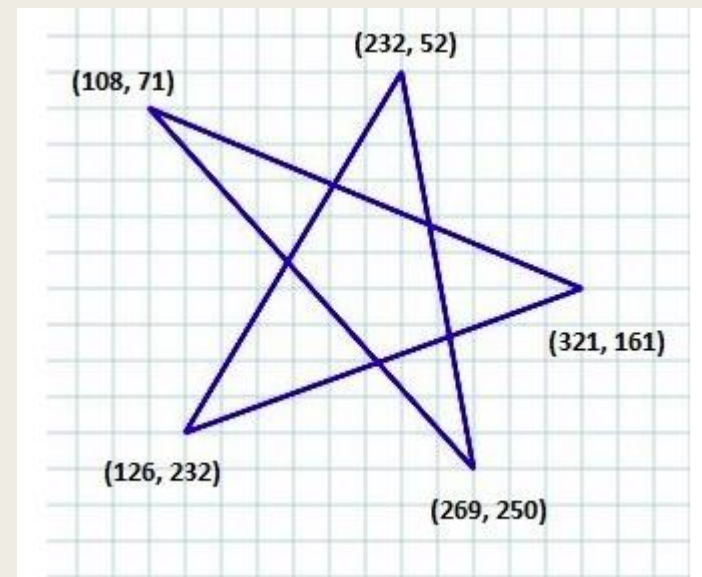
```
p.setStroke(Color.BLACK);
```

```
p.setStrokeWidth(3.0);
```



Классы PathElement

- LineTo
- HLineTo
- VLineTo
- QuadCurveTo
- CubicCurveTo
- ArcTo



Другие классы

- SVGPath траектория на основе строки в формате SVG

```
SVGPath svg3 = new SVGPath();
```

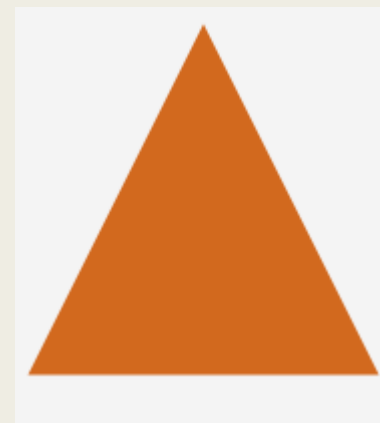
```
svg3.setContent("M 100 100 L 50 200 L 150 200 Z");
```

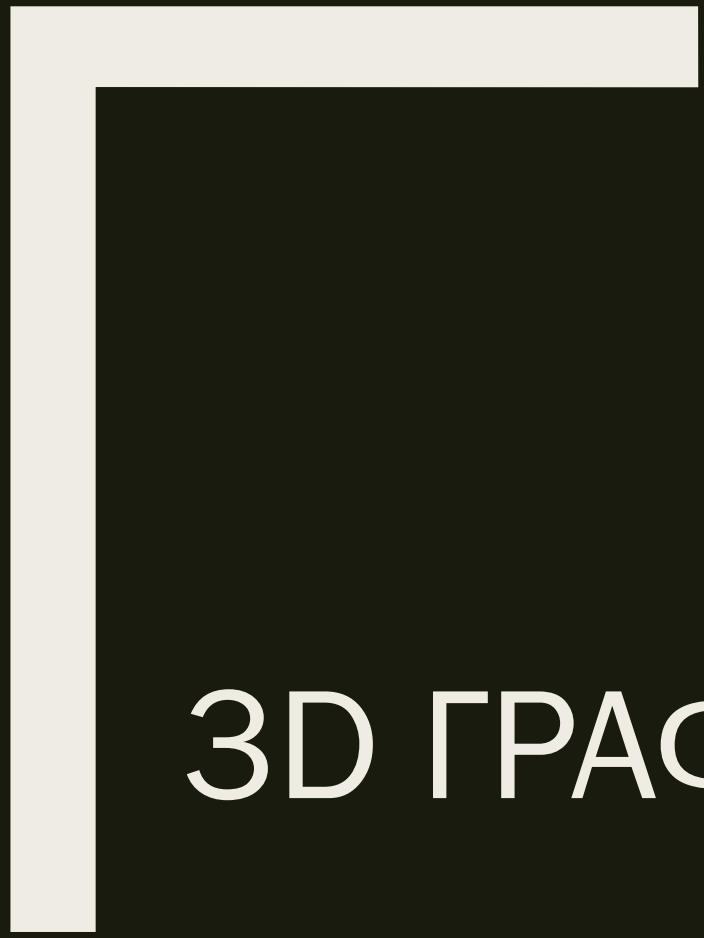
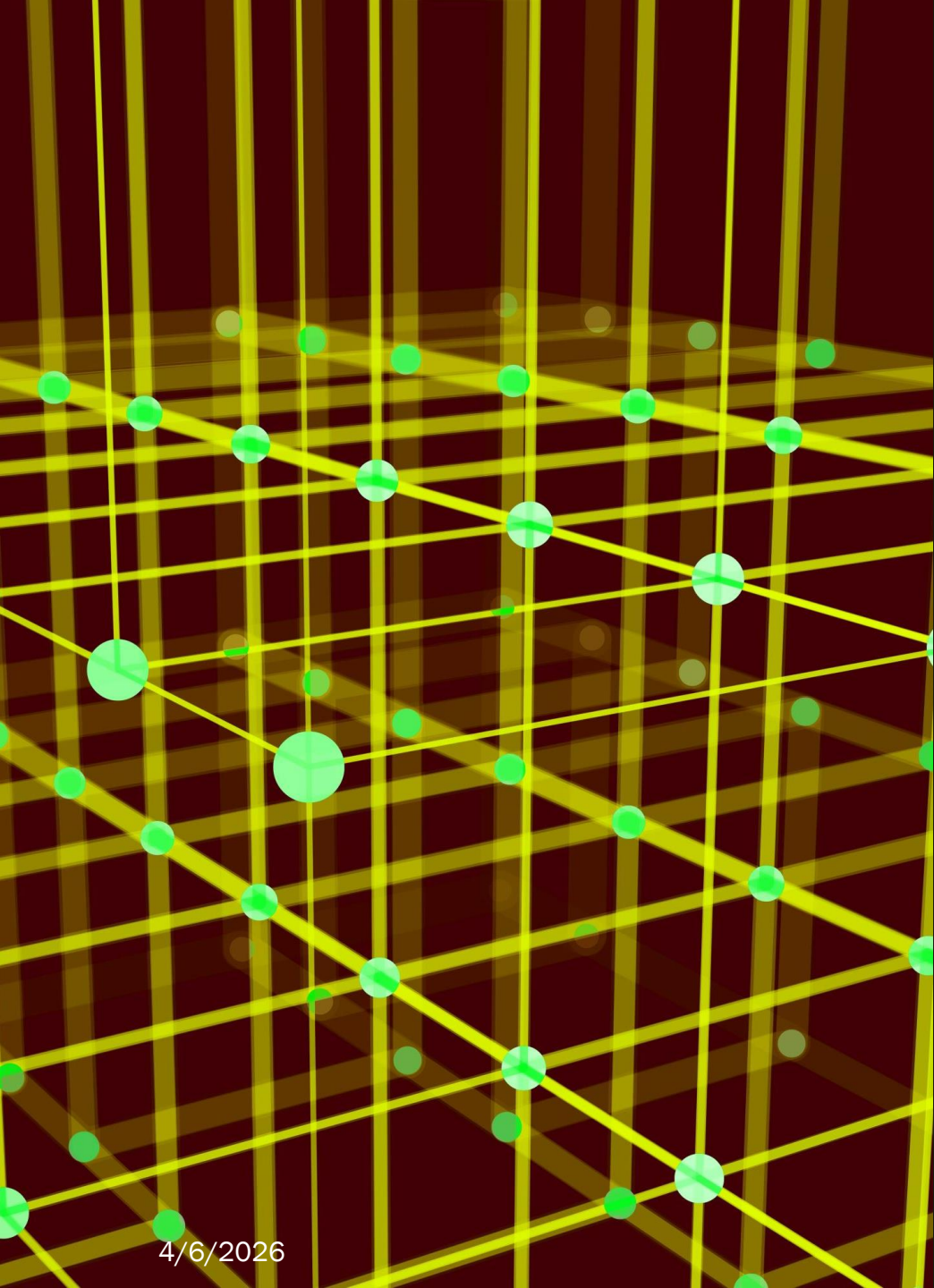
```
svg3.setFill(Color.CHOCOLATE);
```

```
svg3.relocate(250.0, 10.0);
```

www.w3.org/TR/SVG/paths.html

PathDemo.java и Paint.java -примеры





3D ГРАФИКА

Классы

- `javafx.scene.shape.Shape3D`
 - `javafx.scene.shape.Box`
 - `javafx.scene.shape.Cylinder`
 - `javafx.scene.shape.MeshView`
 - `javafx.scene.shape.Sphere`

Классы

- Point3D
- Shape3D – абстрактный базовый
- Box
- Cylinder
- Sphere
- MeshView - 3D фигура произвольной формы

Примеры – Shape3d.java

Shape3D

- Является предком для всех 3D-фигур, обеспечивая общие свойства
 - *Материал определяет внешний вид объекта (цвет, текстура), задается методом `setMaterial()`*
 - *Свойства модели отрисовки, определяющие способ отображения геометрии, задается методом `setDrawMode()`*
 - *Свойства отображения граней, определяющие, какие грани 3D-объекта (передние, задние или оба) должны отрисовываться, что важно для оптимизации рендеринга, задается методом `setCullFace()`*

SubScene

- Если на сцене должны присутствовать и двумерные и трехмерные объекты используется объект SubScene
- Это контейнер, позволяющий встроить одну сцену (сцену в сцене) внутри другой, предоставляя изолированное пространство для 2D или 3D графики
- Управление свойствами трехмерного пространства

Для сцены можно определить наличие буфера глубины

Конструктор позволяет определить наличие буфера глубины и уровень сглаживания

```
SubScene sb = new SubScene( root, 500, 300, true, SceneAntialiasing.BALANCED);  
sb.setCamera(camera);
```

Пример SubScn.java

Камера

Для отображение объектов на сцене используется камера

Классы

- Camera - абстрактный
- ParallelCamera
- PerspectiveCamera

Пример CameraEx.java

Источник света

Классы

- LightBase - абстрактный
- AmbientLight – окружающий свет
- PointLight – точечный источник

Пример Light.java