

## Теоретический материал для индивидуального задания №1

### 1) Java 2D API

Java 2D API становится доступным, начиная с Java 1.2. Возможности этого API включают в себя:

- Класс `Graphics2D`, подкласс `Graphics`, определяющий добавочные графические примитивы и атрибуты.
- Интерфейс `Shape`, который Java 2D использует для определения многих графических примитивов и других операций. Пакет `java.awt.geom` содержит множество полезных реализаций `Shape`.
- Интерфейс `Stroke`, описывающий, как линии проводятся (или рисуются, например, широкой кистью). Класс `BasicStroke` реализует этот интерфейс и поддерживает рисование широких и пунктирных линий.
- Интерфейс `Paint`, описывающий заливку фигур. Его реализация `GradientPaint` позволяет произвести заливку цветовым градиентом, а `TexturePaint` поддерживает заливку «плиткой» (tile, фигура заполняется размноженным изображением). Также, класс `Color` реализует интерфейс `Paint`, который позволяет заливать фигуры сплошным цветом.
- Интерфейс `Composite`, определяющий, как цвета рисунка взаимодействуют с цветами поверхности рисования. Класс `AlphaComposite` позволяет комбинировать цвета на основе уровня прозрачности (alpha transparency). Класс `Color` также содержит новые конструкторы и методы, поддерживающие прозрачные цвета.
- Класс `RenderingHints`, позволяющий приложению запрашивать специальные типы рисования, такие как сглаженное (antialiased) рисование. Сглаживание использует прозрачные цвета и композицию цветов для сглаживания краев букв и других фигур, предупреждая возникновение «ступенек» (jaggies).
- Класс `java.awt.geom.AffineTransform`, осуществляющий трансформации систем координат и позволяющий выполнять перемещение, масштабирование, вращение и перенос фигур (и целых координатных систем). Поскольку `AffineTransform` допускает преобразование систем координат, включая масштабирование, Java 2D API проводит различие между системой координат устройства (то есть пространством устройства) и (возможно) трансформированной системой координат пользователя (то есть пространством пользователя). Координаты в пространстве пользователя могут задаваться значениями типа `float` или `double`; графика, выполняемая Java 2D, не зависит от разрешения устройства.

Кроме того, ваши приложения не ограничены фиксированным набором логических шрифтов; можно применять любой установленный в системе шрифт. Наконец, Java 2D включает удобный в использовании API обработки изображений.

Самостоятельно рассмотреть следующие описание классов/интерфейсов в документации JDK: `Graphics2D`, `Shape`, `Stroke`, `Paint`, `GradientPaint`, `TexturePaint`, `Composite`, `AlphaComposite`, `RenderingHints`, `AffineTransform`.

Дополнительный материал в книге \_Books\corejava2\_1.djvu стр.323-348

## **2) Демонстрационные примеры (в папке Примеры)**

Для запуска примера используйте команду `go.cmd` в каталоге примера (требуется `java.exe`, возможно нужно скорректировать переменную `PATH`)

### 1 Рисование и заливка фигур

Показывает, как можно создать, нарисовать и закрасить различные объекты `Shape`. Фигуры определяются с помощью различных классов, в основном из пакета `java.awt.geom`, как объекты `Shape`. Создание собственного потомка класса `Shape` реализовано в классе `Spiral`.

### 2 Трансформации

Класс `AffineTransform` может трансформировать фигуры (или систему координат), перемещая, масштабируя, вращая и искривляя их, а также применяя произвольные сочетания всех этих преобразований.

Аффинная (affine) трансформация - это линейная трансформация, обладающая двумя важными свойствами: все прямые линии остаются прямыми, а все параллельные линии остаются параллельными. Класс `AffineTransform` может выполнить множество самых разных трансформаций, но не может производить нелинейные эффекты, например искажение изображения, которое происходит при рассматривании рисунка через толстую линзу. Эффекты такого рода могут, достигаться только при использовании техники обработки изображений.

Пример иллюстрирует действие трансформаций координат различных типов на одну фигуру, нарисованную несколько раз.

### 3 Задание стилей линий при помощи класса `BasicStroke`

Пример показывает, как применять `BasicStroke` для рисования толстых линий с различными стилями концевых точек и соединений, а также для рисования шаблонных линий.

### 4 Рисование линий и заливка

Класс `BasicStroke` является реализацией интерфейса `Stroke`. Этот интерфейс в Java 2D отвечает за способ проведения (`draw`) или рисования (`stroke`) линий. Заливка цветом произвольных фигур - это фундаментальная графическая операция, определяемая в Java 2D. Интерфейс `Stroke` определяет API, при помощи которого операция рисования линии преобразуется в операцию заливки области.

### 5 Отчёт

Пример отчёта по результатам решения задания.

**Литература**

1. Хабибуллин И. Ш. Java 7. — СПб.: БХВ-Петербург, 2012
2. Г. Шилдт. Java . Полное руководство, 8-е издание, М.: ООО И.Д. Вильямс, 2012
3. Кей С. Хорстман. Java2 Основы. Том 1. С.-Петербург. 2007 (\_Books\corejava2\_1.djvu)
4. Кей С. Хорстман. Java2 Тонкости программирования. Том 2. С.-Петербург. 2007 (\_Books\corejava2\_1.djvu)
5. <http://docs.oracle.com/javase/8/docs/>