

Grafika 2D

Platformy Technologiczne

Michał Wójcik

Katedra Architektury Systemów Komputerowych
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

10 marca 2011

Java 2D API:

- rysowanie linii i figur,
- wypełnianie figur kolorami, gradientami i teksturami,
- rysowanie tekstu,
- rysowanie obrazków, filtry,
- kompozycja i transformacje.

Graphics — bazowa klasa abstrakcyjna dla wszystkich kontekstów graficznych umożliwiającą rysowanie na komponentach.

Graphics2D — rozszerzenie dostarczające bardziej zaawansowanej kontroli nad rysowaniem:

- rysowanie i wypełnianie:
 - tekstu: `drawText()`,
 - obrazów: `drawImage()`,
 - kształtów geometrycznych: `drawLine()`, `drawArc()`, `drawRect()`, `drawOval()`, `drawPolygon()`,
 - `fill()`;
- konfiguracja atrybutów:
 - `setFont()`, `setColor()`, `setStroke()`, `setPaint()`.

Współrzędne:

- user space:
 - niezależna od urządzenia przestrzeń wykorzystywana przez aplikację,
 - punkt (0,0) w lewym górnym rogu,
 - współrzędne wyrażone w postaci liczb całkowitych,
 - wspiera wartości zmiennoprzecinkowe, również podwójnej precyzji;
- device space — system współrzędnych zależny od urządzenia, konwersja automatyczne podczas renderowania.

Podstawowe kształty:

- Point2D,
- Line2D,
- krzywe:
 - QuadCurve2D,
 - CubicCurve2D,
- Rectangle2D,
- RoundedRectangle2D,
- Ellipse2D,
- Arc2D,
- podklasy:
 - Name2D.Name — publicznie pola typu int,
 - Name2D.Double — publiczne pola typu double,
 - Name2D.Float — publiczne pola typu float.

`GeneralPath` — implementuje interfejs `Shape`, reprezentuje geometryczną ścieżkę stworzoną z linii i krzywych:

- `new GeneralPath()`,
- `moveTo()`,
- `lineTo()`,
- `quadTo()`,
- `curveTo()`,
- `closePath()`,
- `append()`.

Klasy obsługujące obrazki:

- `Image` — abstrakcyjna klasa nadrzędna przedstawiająca obraz jako tablicę pikseli,
- `BufferedImage` — klasa pozwalająca na operacje na obrazku, można tworzyć instancje.

`ImageIO` — wczytywanie i zapisywanie obrazów:

- wsparcie dla: GIF, PNG, JPEG, BMP, WBMP,
- rozszerzalność o inne formaty (dostępne pluginy dla TIFF i JPEG 2000),
- `BufferedImage read(File file)`,
- `boolean write(RenderedImage img, String format, File output)`.

- Obserwator (Observer) — nasłuchuje na zmiany w innym obiekcie, implementuje interfejs `Observer`:
 - `void update(Observable o, Object arg)`.
- Obiekt obserwowany (Observable) — powiadamiania inne obiekty o zmianie swojego stanu, dziedziczy po klasie `Observable`:
 - `setChanged()` — informacja że nastąpiły zmiany,
 - `notifyObservers()` — informacja do obserwatorów,
 - `notifyObservers(Object obj)`.
- Oddzielenie logiki od wyświetlania, osobne moduły do działania na danych i wyświetlania.