

Window-Klasse

Benötigt wird die Library der Window-Klasse (ist z.B. im Projekt U05 enthalten). Die JRE enthält nicht die Window-Klasse, mit der wir arbeiten.

Um die Klasse zu nutzen:

```
import gui.Window;

Window window = new Window("Empty", 500, 300);
```

Alle Window Methoden:

<code>open()</code>	Fenster wird geöffnet, d.h. sichtbar
<code>close()</code>	Fenster wird geschlossen, d.h. unsichtbar
<code>waitUntilClosed()</code>	Programm wartet, bis Benutzer das Fenster schliesst (z.B. durch den ● oder X klicken)
<code>isOpen()</code>	Gibt true zurück, falls Fenster gerade offen ist, sonst false

`fillRect(double x, double y, double width, double height)`
 Zeichnet ein (ausgefülltes) Rechteck der Grösse `width` × `height`, welches die linke obere Ecke beim Punkt (`x`, `y`) hat

`fillCircle(double x, double y, double radius)`
 Zeichnet einen (ausgefüllten) Kreis mit Radius `radius`, welches den Mittelpunkt beim Punkt (`x`, `y`) hat

`fillOval(double x, double y, double width, double height)`
 ...

`refresh()`
 Zeigt die aktuelle Zeichnung im Fenster an.

`refresh(int waitTime)`
 Zeigt die aktuelle Zeichnung im Fenster an. Um ein konstantes Zeitintervall zwischen mehreren Refreshs zu erreichen, zeigt diese Methode die Änderungen erst `waitTime` Millisekunden nach dem *letzten Aufruf* von `refresh()` an.

`refreshAndClear(int waitTime)`
 Zeigt wie `refresh(waitTime)` die aktuelle Zeichnung an, aber leert die Zeichnungsfläche danach für ein neues Bild.

`isKeyPressed(String keyName)`

Gibt **true** zurück, falls gegebene Taste im Moment gerade gedrückt wird, **false** andernfalls

`isLeftMouseButtonPressed() / isRightMouseButtonPressed()`

Gibt **true** zurück, falls die linke/rechte Maustaste im Moment gerade gedrückt wird, **false** andernfalls

`wasKeyTyped(String keyName) / was...MouseButtonClicked()`

Gibt **true** zurück, falls die gegebene Taste gerade «getippt»/«geklickt» wurde, **false** andernfalls

`getMouseX() / getMouseY()`

Gibt die aktuelle X-/Y-Koordinate des Mauszeigers (Cursors) zurück

137

`drawRect(x, y, width, height)`

Zeichnet die Kontur eines Rechtecks

`drawCircle(x, y, radius)`

`drawLine(x1, y1, x2, y2)`

`drawString(string, x, y)`

Zeichnet den gegebenen String so, dass sich die Grundlinie des ersten Zeichens bei (x, y) befindet

`drawImage(path, x, y)`

Zeichnet das Bild, das sich in der Datei bei `path` befindet, mit der linken oberen Ecke bei (x, y)

`drawImageCentered(path, x, y)`

`setColor(red, green, blue)`

Setzt die Farbe für die folgenden Zeichenbefehle auf den **RGB-Wert** gegeben durch **red, green, blue**

`setStrokeWidth(width)`

Setzt die Strichdicke für die folgenden **draw**-Zeichenbefehle

`setFontSize(fontSize)`

Setzt die Schriftgrösse für die folgenden **drawString**-Befehle

...

139