

Dokumentation der benötigten Klassen (Neu hinzugekommenes in blauer Schrift)

<code>import pm.gamewindow.*;</code>	Importiert das GameWindow-Package
<code>import java.awt.*;</code>	Importiert das AWT-Package (für die Klasse Color)
<code>import java.awt.event.*;</code>	Importiert das Event-Package (für KeyEvent)

Klasse GameWindow

Konstruktor (Beispiel): <code>window = new GameWindow(50, 50, 800, 600, "Spiel");</code> Erzeugt ein GameWindow an Position x=50 y=50 auf dem Bildschirm mit Breite 800 und Höhe 600.	
<code>void drawLine(int px1, int py1, int px2, int py2, Color pc)</code> Zeichnet eine Linie von px1 / py1 nach px2 / py2 in der Farbe pc.	
<code>void clear()</code> <code>void clear(Color pc)</code>	Übermalt das Fenster in weiß, bzw. mit der Farbe pc
<code>void paintFrame()</code>	Überträgt das bisher gezeichnete an den Bildschirm
<code>int getWidth()</code> / <code>int getHeight()</code>	Geben die Breite / Höhe des Fensters zurück
<code>int getMouseX()</code> / <code>int getMouseY()</code>	Geben die aktuelle Position des Mauszeigers zurück
<code>boolean mouseButton1()</code> / <code>boolean mouseButton2()</code> Gibt true zurück, wenn die linke (bzw. rechte) Maustaste gedrückt wurde. Die nächsten Aufrufe bleiben dann solange false, bis die Taste wieder gedrückt wird.	
<code>boolean isKeyDown(int pKeyCode)</code> Bsp: <code>if (window.isKeyDown(KeyEvent.VK_RIGHT)) ...</code> Gibt true zurück, falls die durch pKeyCode angegebene Taste gerade gedrückt ist. Für pKeyCode werden Codes der Klasse KeyEvent eingesetzt, unter anderem VK_LEFT, VK_RIGHT, VK_UP, VK_DOWN: Pfeiltasten VK_A ... VK_Z, VK_0 ... VK_9: Buchstaben-Tasten / Nummern 0 bis 9 VK_SPACE, VK_ENTER: Leertaste, Enter	
<code>int mouseWheel()</code> Gibt zurück, wie viele „notches“ sich das Mousrad seit der letzten Abfrage bewegt hat. 0 = keine Bewegung, negativer Wert = Abwärts-, positiver Wert = Aufwärtsdrehung	

Klasse Sprite

Konstruktor (Beispiel): <code>sp = new Sprite("./images/ghost.png");</code>	
<code>void setPosition(double px, double py)</code>	Setzt die Position des Sprites im GameWindow (linke obere Ecke des Sprite-Bildes)
<code>void setSpeed(double ps)</code>	Setzt die Geschwindigkeit (in Pixeln)
<code>void schneller(double ps)</code> <code>void langsamer(double ps)</code>	Erhöht bzw. verringert die Geschwindigkeit
<code>void setRichtung(double pr)</code>	Setzt die Richtung in Grad (0° entspr. x-Achse)
<code>void dreheLinks(double pr)</code> <code>void dreheRechts(double pr)</code>	Dreht die Richtung nach links bzw. rechts
<code>void bewege()</code>	Ändert die Position um „Speed“ in „Richtung“.
<code>void draw(GameWindow pwin)</code>	Zeichnet das Bild des Sprites an der aktuellen Position x / y auf das GameWindow pwin.

Aufgabe

Nutze für die Aufgaben die bereitgestellte Vorlage.

- a) Implementiere die Methode **aufgabe1()**:
Ein Sprite wird zu Beginn in die Mitte des Fensters gesetzt. Jedesmal, wenn die linke **Maustaste** gedrückt wird, springt der Sprite an die Position des Mauszeigers.
- b) Implementiere die Methode **aufgabe2()**:
Ein Sprite wird zu Beginn in die Mitte des Fensters gesetzt.
Wenn die **Taste** „Pfeil links“ gedrückt wird, bewegt er sich ein Stück nach links, und entsprechend mit den anderen Pfeiltasten in die Richtungen rechts, oben und unten.
(Solange die Taste gedrückt bleibt, bewegt sich der Sprite dann flüssig weiter.)
- c) Implementiere die Methode **aufgabe3()**:
Ein Sprite wird zu Beginn in die Mitte des Fensters gesetzt.
Wenn auf der Tastatur die **Taste** „Pfeil oben“ gedrückt wird, wird seine Geschwindigkeit etwas erhöht. Er bewegt sich dann in seiner aktuellen Richtung weiter (ohne zu stoppen).
Dadurch wird er schneller, je länger die Taste gedrückt bleibt.
Bei Taste „Pfeil unten“ wird er langsamer. Wenn die Tasten „Pfeil links“ bzw. „rechts“ gedrückt werden, ändert sich seine Richtung leicht nach links bzw. rechts.
- d) Implementiere die Methode **aufgabe4()**:
Das Fenster füllt sich zu Beginn mit der Farbe grau (rot, grün und blau 50%).
Wenn das **Mausrad** nach oben gedreht wird, wird die gesamte Fläche des Fensters heller, wenn es nach unten gedreht wird, dunkler.
Achte darauf, dass die Farbwerte nicht kleiner als 0 bzw. größer als 1 werden.
Hinweis: die Methode `mouseWheel()` sollte nur einmal pro Wiederholung abgefragt werden.
Speichere den Rückgabewert der Methode also in einer Variable, um ihn 2x zu verwenden.
- e) Implementiere die Methode **aufgabe5()**:
Von der Mitte des Fensters ausgehend: Wenn die linke **Maustaste** gedrückt wird, wird eine Linie von der Mitte zur Position des Mauszeigers gezeichnet. Von dort aus geht es dann weiter, bei jedem Klick wird vom letzten Punkt aus eine Linie weiter zum Mauszeiger gezogen.