

Objektorientiertes Programmieren mit GPanel-Grafik

Das Projekt MeinPark, III

Aufgabe 9: Bewegung

Nun soll etwas Bewegung ins Spiel kommen, indem sich die Sonne zufällig auf dem GPanel bewegt. Dazu muss sie einfach nur gelöscht (d.h. erneut mit der Hintergrundfarbe gezeichnet) und nahe der alten Position an anderer Stelle neu gezeichnet werden.

Durch das permanente Zeichnen (abwechselnd in weiß und gelb) entstehen jedoch störende Linien, was durch kleine Pausen zwischen den Bewegungen verhindert werden kann. Der dazu notwendige Befehl lautet: `GPanel.delay(10);` und bewirkt, dass das Programm für 10 Millisekunden unterbrochen wird.

Exkurs: Statische Methoden

Fällt dir etwas auf: Vor dem eigentlichen Befehl `delay` steht dieses Mal nicht der Name des GPanel-Objekts (wie z.B. bei dem Befehl `gp.move(...)`) sondern `GPanel`, also der Name der Klasse selbst. Was hat es damit auf sich?

Methoden können zu Objekten oder Klassen gehören. Wenn Methoden zu Klassen gehören, schreiben wir

```
Klassenname.methodenname(parameter);
```

um die Methode aufzurufen. Wenn eine Methode zu einem Objekt gehört, schreiben wir

```
objekt.methodenname(parameter);
```

um die Methode aufzurufen. Beide Arten von Methoden sind in einer Klasse definiert. Die Methodensignatur verrät uns, ob eine gegebene Methode zu den Objekten dieser Klasse oder der Klasse selbst gehört.

Statische Methoden:

Methoden, die sich nicht auf ein bestimmtes Objekt beziehen, sondern zu der Klasse selbst gehören, heißen statische Methoden und werden am Anfang der Methodensignatur durch das Schlüsselwort `static` gekennzeichnet.

In Aufgabe 2 hast du am Beispiel der Klasse `BaseTimer` gelernt, wie du mit der Dokumentation der `aplu`-Klassen umgehen musst. Suche dort einmal die Methode `delay` der Klasse `GPanel` heraus. Erkennst du, dass die Signatur so lautet?

```
static void delay() (int time)
```

Objektorientiertes Programmieren mit GPanel-Grafik

Das Projekt MeinPark, III

Daran können wir ablesen, dass die Methode `delay` eine statische Methode ist und wir den Namen der Klasse (`GPanel`) vor dem Punkt in dem Methodenaufruf schreiben müssen.

(Ein weiteres Beispiel für eine statische Methode ist die Methode `random` der Klasse `Math`: Sie ist eine Klassenmethode, bezieht sich nicht auf ein bestimmtes Objekt und wird daher aufgerufen, indem der Klassenname vorangestellt wird: `Math.random()`. Du kannst das übrigens überprüfen, indem du unter

<https://docs.oracle.com/javase/7/docs/api/>

In der Klasse `Math` die Methode `random` heraussuchst).

- a) Füge deiner Klasse `Sonne` eine Methode `private void verschwinde()` hinzu. Sie soll die Sonne einfach in deiner Hintergrundfarbe zeichnen.
- b) Füge deiner Klasse `Sonne` eine Methode `public void bewegeZufaellig()` hinzu. Sie soll wie oben beschrieben unter Verwendung der Methode aus a) bewirken, dass sich die Sonne zufällig um 10 Pixel nach links, rechts, oben oder unten bewegt. Baue nach dieser Bewegung eine Pause von einigen Millisekunden ein, um störendes Flackern zu verhindern.
- c) Verändere dein Hauptprogramm `MeinPark` derart, dass Sonne und Blumen direkt von Beginn an sichtbar sind und sich die Sonne vom Start des Programms an so lange bewegt, bis durch einen Mausklick das Programm beendet wird.
- d) Durch die zufällige Bewegung kann es passieren, dass sich die Sonne aus dem Fenster heraus bewegt. Das sollte natürlich vermieden werden. Verändere dazu deine Methode `bewegeZufaellig()` derart, dass eine Bewegung der Sonne nach oben/unten/links/rechts nur dann ausgeführt wird, wenn der Abstand der Sonne zum oberen/unteren/linken/rechten Bildschirmrand groß genug ist.

Tipp: Die Breite eines GPanel-Objekts namens `gp` erhältst du durch den Befehl

```
gp.getWindow().getWidth()
```

Die Höhe bekommt man mit Hilfe von

```
gp.getWindow().getHeight()
```

Du kannst aber natürlich auch die absoluten Zahlen dafür einsetzen.

- e) Durch die Bewegung der Sonne werden im Weg stehende Blumen offenbar gelöscht. Dies sollst du verhindern, indem du dein Hauptprogramm derart erweiterst, dass alle Blumen immer wieder neu an ihre Position gezeichnet werden.