

13.4 Übungsblatt 4

Dieses Blatt beginnt wieder mit einer Theoriefrage, die bitte per Textdatei in einem Package namens `blatt04` beantwortet werden soll.

Weiter geht es mit einigen Grafik-Programmieraufgaben, bei denen insbesondere das Arbeiten mit Instanzmethoden geübt werden soll.

Aufgabe 4.1: Lesen

Lesen Sie sich zunächst die folgenden Kapitel im [Javabuch] durch:

- 3.4.1 Kompilieren
- 4.1 Lokale Variablen, Ausdrücke und Schleifen
- 6.3.2.1 Methoden
- 12.4 Benutzung von Paketen

Sie sollten Fragen zu diesen Inhalten beantworten können.

13.4.1 Der Java Kompiliervorgang

Aufgabe 4.2*: Java-Programmierfehler selbst erkennen

Finden Sie bitte in folgenden Java-Programmstücken den jeweiligen Fehler.

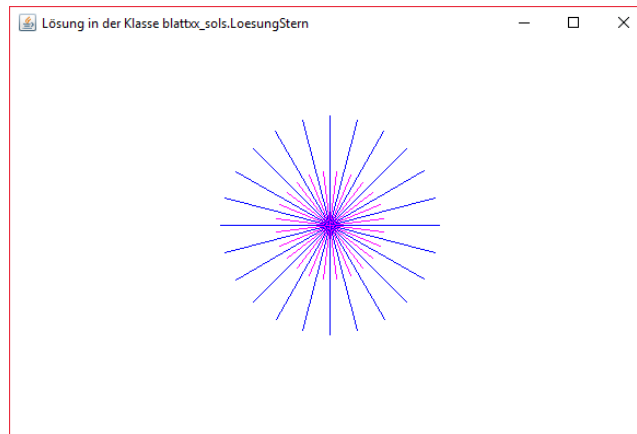
Java-Programmstück	Grund
<code>for(int i=0, i<10, i=i+1);</code>	
<code>int k = 0; System.out.println(1/k);</code>	
<code>int i=0; (i < 1) {i=i+1;}</code>	
<code>unsigned int i=0; while(i < 1) {i=i+1;}</code>	
<code>long i=1; while(i > 0) {i=i+1;}</code>	

13.4.2 Weitere Grafikprogramme

OK, Weihnachten ist noch weit weg, aber trotzdem kann man sich ja schon mal um ein selbstgestaltetes Geschenkpapier ein paar Gedanken machen ...

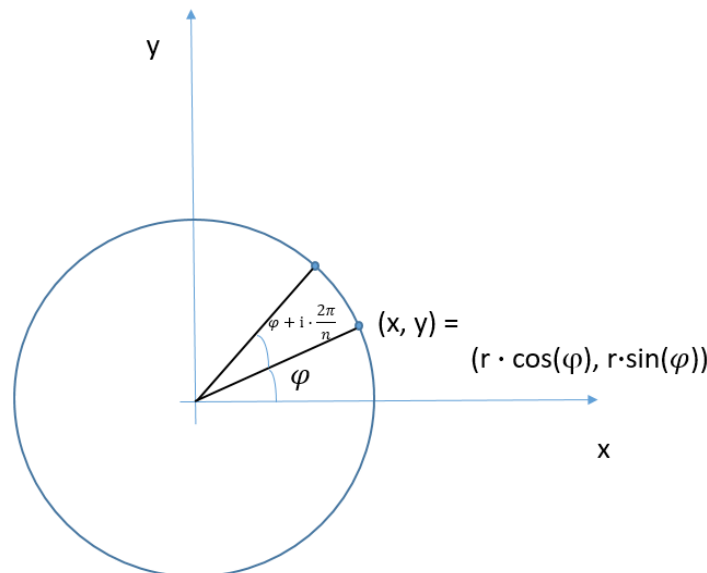
Aufgabe 4.3: Einen Stern malen

In dieser Aufgabe soll ein einfacher Stern in die Mitte des Fensters gezeichnet werden:



Die zu ändernde Klasse `LoesungStern` ist im Ordner `blatt04` in ILIAS zu finden. In dieser Klasse soll die Methode `meinStern()` geändert werden, um einen Stern zu zeichnen.

Dazu etwas Hilfestellung:



Die x - und y -Koordinate irgendeines Punktes auf einem Kreis mit dem Radius r erhält man, indem man

$$x = r \cdot \cos(\varphi)$$

und

$$y = r \cdot \sin(\varphi)$$

setzt, jeweils mit φ einer beliebigen Zahl zwischen 0 und $2 \cdot \pi$.

Wenn man jetzt n Punkte auf einem Kreis haben will, kann man in einer `for`-Schleife eine Zählvariable i von 0 bis $n-1$ laufen lassen und den laufenden Winkel so berechnen:

$$\varphi_i = i \cdot (2 \cdot \pi) / n.$$

Damit der Stern in der Mitte des Fensters ist, muss man dazu (und zum Mittelpunkt $(0, 0)$ des Sternes von dem alle Strahlen des Sternes ausgehen) noch jeweils `this.getWidth() / 2`. bzw. `this.getHeight() / 2`. addieren.

Diese Aufgabe ist ein schönes Beispiel für die Verwendung einer Herangehensweise, die sich *Schrittweise Verfeinerung* nennt. Man fängt mit einer groben Lösung an, testet diese und verfeinert erst dann diese Lösung, wenn der Test erfolgreich war. Für die gegebene Aufgabe kann das so aussehen:

1. Eine for -Schleife schreiben, die für $n = 8$ die ϕ_i auf der Konsole ausgibt. Prüfen ob die Werte "vernünftig" sind. Die Kreiszahl π können Sie über `Math.PI` erhalten.
2. Dann mit `Math.cos(phi)` und `Math.sin(phi)` die n x- und y-Koordinaten berechnen, ausgeben und prüfen.
3. Jetzt erst die Linien malen.
4. Den Stern in die Mitte schieben.
5. Bei den ungeraden Strahlen des Sternes den Radius r etwas kleiner machen, z.B. $r/2$.
6. Farben der Strahlen verändern.

Aufgabe 4.4: Eine Instanzmethode schreiben, die den Stern malt

Wenn wir so weit sind, sollten wir das Malen des Sternes in einer neuen Methode parametrisieren, um Sterne beliebig groß und an einem beliebigen Ort malen zu können.

Schreiben Sie also bitte eine Methode mit folgender Signatur:

```
public void zeichneEinenStern(Graphics g, int midX, int midY, int r)
```

`midX` und `midY` sind dabei der Mittelpunkt des Sternes, r dessen Radius.

Übernehmen Sie den Rumpf der `meinStern()` -Methode aus der vorherigen Aufgabe.

Testen Sie die neue `zeichneEinenStern()` -Methode in der `meineZufallsSterne` -Methode mit verschiedenen Radien und Mittelpunkten.

Dieses Auslagern einer mehrfach benötigten Funktionalität (hier das Malen des Sternes) in eine neue Methode nennt man *Refactoring*. Probieren Sie doch bitte, ob Sie die komplett markierte for -Schleife der alten `draw`-Methode unter Eclipse mit Rechtsklick -> Refactor -> Extract Method nicht sehr bequem auslagern können. Oder lassen Sie sich das zeigen ...

Aufgabe 4.5*: Das Geschenkpapier ...

Jetzt sind wir in der Lage, beliebig viele, verschieden große Sterne zu malen.

Ändern Sie dazu bitte die in ILIAS gegebene Klasse `LoesungZufallsSterne`.

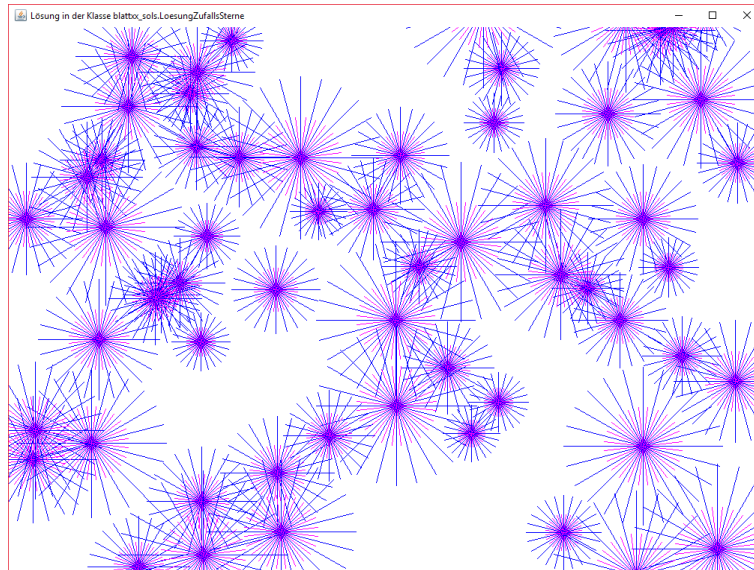
Nutzen Sie für die zufällige Platzierung der Sterne die `Math.random()`-Funktion, die zufällige Werte liefert, die größer oder gleich 0,0 sind und die kleiner als 1,0 sind. Eine zufällige x-Koordinate kann man also so berechnen:

```
double x = Math.random() * this.getWidth();
```

Die ganzzahlige Variante der zufälligen x-Koordinate erhält man durch einen Cast:

```
int ix = (int) (Math.random() * this.getWidth());
```

Das Ergebnis sollte (in etwa) so aussehen:



Aufgabe 4.6: Zusatz: etwas ausprobieren ...

Sie können sehr gerne natürlich noch etwas mit den Farben, den Strahlen der Sterne oder auch den nicht mehr zufällig gewählten Positionen der Sterne wie im folgenden Beispiel arbeiten und Ihrer Phantasie freien Lauf zu lassen

