

1

Software Entwicklung Probetest

Lesen Sie sich die Aufgabenstellungen sorgfältig durch und achten Sie auf genaue Ausgaben.
Bitte am Beginn des Programms Namen implementieren.

Für diesen Probetest besteht wie immer keine pflichtige Abgabe. Wird sich trotzdem für eine Einreichung des Ergebnisses entschieden, so bekommt man auch ein Feedback!

Unter folgendem Link können Sie auf die Files zugreifen:

https://github.com/J JulianWarlischek/SEW_ProbeTest-01.git

Benotung:

Aufgabe	Punkte
<i>Flower Processing</i>	25 Punkte
<i>SortAndSearch</i>	15 Punkte

Fehler:

Aufgabe	Fehler
<i>Flower Processing</i>	
<i>SortAndSearch</i>	

Note:

Viel Erfolg 😊

1

Software Entwicklung Probetest

Aufgabe 1: Flower Processing

In diesem Programm zeichnen wir eine Blume in Processing. Ähnlich wie der Baum in den vergangenen Unterrichtseinheiten?

Ja! Dieser Algorithmus funktioniert ähnlich. Die Blüten der Blume werden mittels Rekursion gezeichnet. Der Stängel lässt sich mithilfe eines bestimmten Kommandos zeichnen. Wichtige bzw. unbekannte Kommandos funktionieren folgendermaßen:

fullScreen() – Aufruf in der **SETTINGS** Methode. Wichtig ist, dass das **size()** Kommando **entfernt** wird. Mittels „**Esc**“ kann man das Fullscreen-Fenster wieder verlassen.

curve() – Zeichnen des Stängels. Diese Methode benötigt **8 Parameter**:

(Kontrollpunkt x1, Kontrollpunkt y1, Punkt x2, Punkt y2, Punkt x3, Punkt y3, Kontrollpunkt x4, Kontrollpunkt y4)

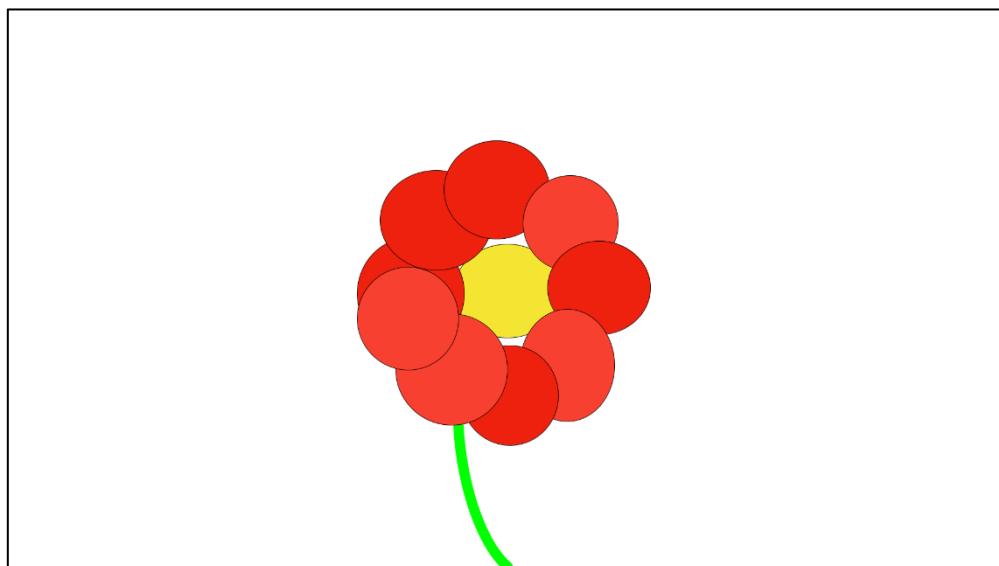
Ellipse () – Zeichnen der Blüten. Diese Methode benötigt **4 Parameter**:

(Punkt x, Punkt y, Durchmesser 1, Durchmesser 2)

Die restlichen Codefragmente sollten bereits bekannt sein.

Hinweis: Stellen Sie die Bildwiederholungsrate auf 1, um die Anzahl der neuen Blumengenerierung zu vermindern.

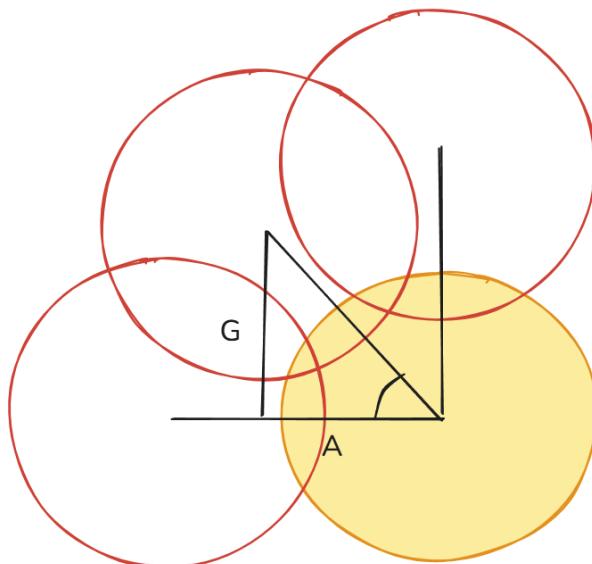
An folgender Ausgabe können Sie sich orientieren:



1

Software Entwicklung Probetest

Folgende mathematischen Zusammenhänge können hilfreich sein:



$$\sin(\text{angle}) = (G / H)$$

$$\cos(\text{angle}) = (A / H)$$

Aufgabe 2: SortAndSearch

In diesem Programm gibt der Benutzer Zahlen im Intervall [1;25] ein. Dabei beträgt das **Maximum 10 Zahlen**. Während der Eingabe kann die Eingabe jedoch jederzeit mit der Eingabe einer 0 abgebrochen werden. Wichtig ist auch, dass **jede Zahl eindeutig** ist und nicht doppelt vorkommt. Gehe dazu folgendermaßen vor:

1. **Eingabe des Benutzers (z.B. 9)**
2. **Suchen der Zahl im Array mittels binärer Suche (Prüfen auf Duplikate)**
3. **Prüfen auf gültigen Bereich**
4. **Alles gültig -> speichern im Array**
5. **Sortiere das Array mithilfe eines Insertion-Sort**

Nach der Eingabe soll das sortierte Array ausgegeben werden (**Nullen sind nicht auszugeben**). Zusätzlich soll der Benutzer erfahren, welche Zahlen **gerade und welche ungerade sind**. Gib jeweils auch die **Anzahl der gerade- bzw. der ungeraden Zahlen** aus. Wichtig ist, dass sie auf die unten angefügte Ausgabe achten und versuchen diese möglichst äquivalent nachzustellen.

1

Software Entwicklung Probetest

Ausgabe:

```
Geben Sie 10 Zahlen im Bereich [1,25] ein (0 für Abbruch der Eingabe):
```

```
Zahl 1: 23
```

```
Zahl 2: 40
```

```
Ungültige Eingabe!
```

```
Zahl 2: 2
```

```
Zahl 3: 23
```

```
Ungültige Eingabe!
```

```
Zahl 3: 5
```

```
Zahl 4: 7
```

```
Zahl 5: 16
```

```
Zahl 6: 13
```

```
Zahl 7: 0
```

```
Sortiert: 2, 5, 7, 13, 16, 23,
```

```
Gerade Zahlen: 2, 16,
```

```
Anzahl: 2
```

```
Ungerade Zahlen: 5, 7, 13, 23,
```

```
Anzahl: 4
```