

Aufgabenblatt 6, Abgabetermin 22.06.2020

Ab diesem Aufgabenblatt benutzen wir die Module `stdraw` und `pygame`.
Installationsanleitungen finden Sie auf den Webseiten zum Buch
<https://introcs.cs.princeton.edu/python/home> unter „To get started“.

Lösen Sie drei Aufgaben. Sie können maximal 15 Punkte erreichen.

Auf dem Kreis mit Radius 1 um den Punkt $(0,0)$ hat der Punkt, den man im Winkel α° von der Linie zwischen $(0,0)$ und $(1,0)$ erreicht, die Koordinaten $(\cos(\frac{\alpha}{180} \cdot \pi), \sin(\frac{\alpha}{180} \cdot \pi))$. Die dabei verwendeten Funktionen gibt es als `math.cos(...)`, `math.sin(...)` und `math.pi` im Modul `math`.

Aufgabe 36: Säulendiagramm

5 Punkte

Stellen Sie die Messergebnisreihen *Jahr Messergebnis*, die das Programm `jahres_mittel.py` liefert, als Säulendiagramme dar. (Die Achsen mit den Beschriftungen müssen Sie nicht darstellen.) Stellen Sie dabei die Werte, die über dem Durchschnitt aller Messergebnisse liegen, mit einer anderen Farbe dar, als die unterdurchschnittlichen Werte.



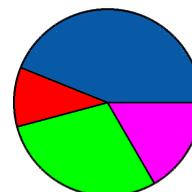
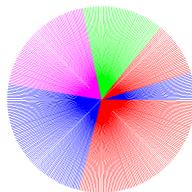
Aufgabe 37: Tortendiagramm

5 Punkte

Schreiben Sie ein Programm, das eine beliebige Anzahl positiver Zahlen einliest und den Anteil jeder Zahl an der Summe aller Zahlen als Tortendiagramm anzeigt.

```
python3 tortendiagramm.py 1 2 3 4 5 6
```

```
python3 tortendiagramm.py 10.2 2.4 6.8 3.9
```



Aufgabe 38: Mit Farben gefüllte konzentrische Kreise

5 Punkte

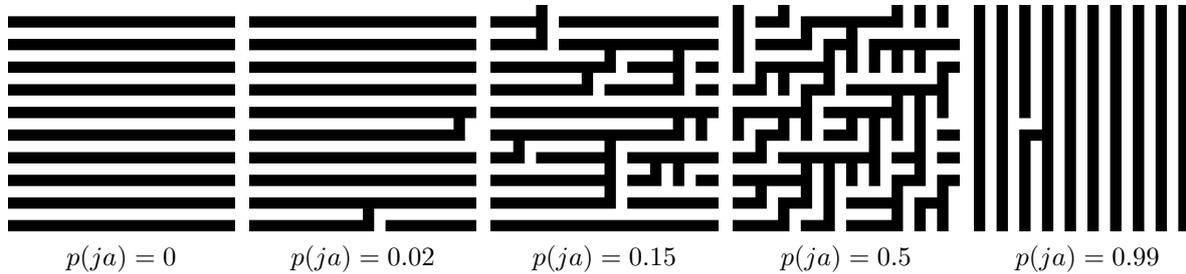
Schreiben Sie ein Programm, das eine Animation eines Kreises produziert, dessen Farbe sich schrittweise ändert. Der Kreis dehnt sich erst in einer Farbe aus. Sobald die ganze Kreisfläche mit der Farbe gefüllt ist, beginnt sich von innen ein Kreis mit einer anderen Farbe auszudehnen. Das wiederholt sich, bis man das Programm abbricht.



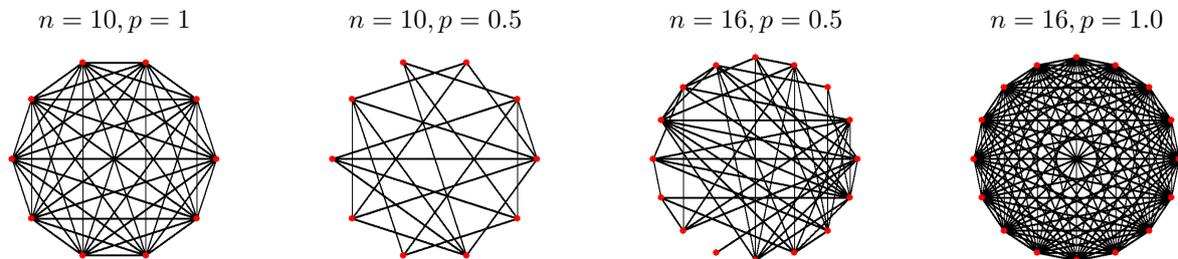
Aufgabe 39: Konkrete Kunst**5 Punkte**

Wolfgang Bosse (geb. 1958) hat bei der Ausstellung *Konkrete Kunst* in der Kunstsammlung Jena (2019) Bilder mit Titeln wie z.B. *Ja-Nein*, $p(\text{ja}) = \frac{1}{8}$ gezeigt. Unten sehen Sie fünf Bilder dieser Art, die ich für unterschiedliche Werte von $p(\text{ja})$ produziert habe. p steht für *Wahrscheinlichkeit*, und $p(\text{ja})$ ist ein Wert aus dem Bereich $0 \dots 1$.

Überlegen Sie, was in den Bildern durch Zufall gesteuert wird (es ist das gleiche „Experiment“ an 10×10 Stellen). Schreiben Sie ein Programm, das einen $p(\text{ja})$ -Wert einliest und ein zufälliges Ja-Nein-Bild mit dieser Wahrscheinlichkeit für „ja“ erzeugt.

**Aufgabe 40: Kreismuster****5 Punkte**

Malen Sie Bilder der folgenden Art.

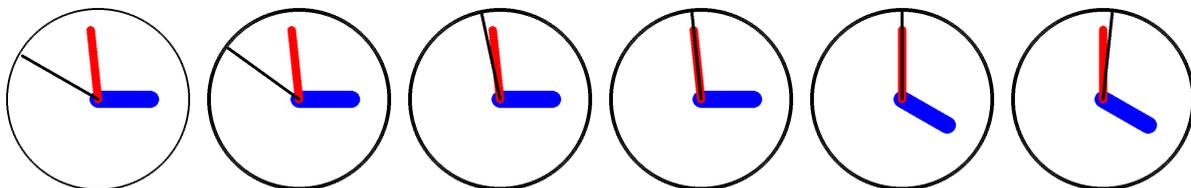


Schreiben Sie dazu ein Programm, das eine `int`-Zahl n und eine `float`-Zahl p aus dem Bereich $0 \dots 1$ einliest. Es soll n auf einem Kreis liegende Punkte zeichnen, die jeweils die gleichen Abstände voneinander haben. Anschließend soll für jedes Paar von Punkten mit der Wahrscheinlichkeit p eine Verbindungslinie gezeichnet werden.

Aufgabe 41: Uhr**5 Punkte**

Programmieren Sie eine Uhr mit Stunden-, Minuten- und Sekundenzeiger. Sie startet bei der eingegebenen Uhrzeit und läuft dann in Sekundenschritten

```
python3 uhr.py 15 59 50
```



Hinweis:

Eine Farbe kann man wie folgt zufällig auswählen.

Zuerst definiert man sich ein Tupel mit Farben (sie stehen auf der nächsten Seite).

```
FARBEN = ( stddraw.BLACK, stddraw.BLUE, ..., stddraw.YELLOW )
```

Dann benutzt man die Funktion `random.choice(...)`, um zufällig ein Element aus einem Array zu wählen.

```
zufallsfarbe = random.choice(FARBEN)
```

| <i>function call</i> | <i>description</i> |
|--|--|
| <i>basic functions for drawing</i> | |
| <code>std draw.line(x0, y0, x1, y1)</code> | draw a line from (x0, y0) to (x1, y1) |
| <code>std draw.point(x, y)</code> | draw a point at (x, y) |
| <code>std draw.show()</code> | show the drawing in the standard drawing window (and wait until it is closed by the user) |
| <i>control functions for setting drawing parameters</i> | |
| <code>std draw.setCanvasSize(w, h)</code> | set the size of the canvas to w-by-h pixels (w and h default to 512) |
| <code>std draw.setXscale(x0, x1)</code> | set the x-range of the canvas to (x0, x1) (x0 defaults to 0 and y0 defaults to 1) |
| <code>std draw.setYscale(y0, y1)</code> | set the y-range of the canvas to (y0, y1) (y0 defaults to 0 and y1 defaults to 1) |
| <code>std draw.setPenRadius(r)</code> | set the pen radius to r (defaults to 0.005) |
| <i>Note: If the pen radius is 0, then points and line widths will be the minimum possible size.</i> | |
| <i>functions for drawing shapes</i> | |
| <code>std draw.circle(x, y, r)</code> | draw a circle of radius r centered at (x, y) |
| <code>std draw.square(x, y, r)</code> | draw a 2r-by-2r square centered at (x, y) |
| <code>std draw.rectangle(x, y, w, h)</code> | draw a w-by-h rectangle with lower-left endpoint (x, y) |
| <code>std draw.polygon(x, y)</code> | draw a polygon that connects (x[i], y[i]) |
| <i>Note: filledCircle(), filledSquare(), filledRectangle(), and filledPolygon() correspond to these and draw filled shapes, not just outlines.</i> | |
| <i>functions for drawing shapes</i> | |
| <code>std draw.text(x, y, s)</code> | draw string s, centered at (x, y) |
| <code>std draw.setPenColor(color)</code> | set the pen color to color (defaults to std draw.BLACK) |
| <code>std draw.setFontFamily(font)</code> | set the font family to font (defaults to 'Helvetica') |
| <code>std draw.setFontSize(size)</code> | set the font size to size (defaults to 12) |
| <i>functions for animation</i> | |
| <code>std draw.clear(color)</code> | clear the background canvas (color every pixel color) |
| <code>std draw.show(t)</code> | show the drawing in the standard drawing window and wait for t milliseconds |

Our std draw module

aus Sedgewick, Wayne, Dondero: Introduction to Programming in Python (Addison-Wesley 2015)

`std draw.save(string)` speichert die Leinwand in der Datei mit dem Namen `string`. Der Dateiname muss mit `.jpg` oder `.png` enden.

Die Farben, die man mit `std draw.FARBE` benutzen kann, sind:

| | | |
|-----------------|------------|--------|
| BLACK | DARK_GRAY | ORANGE |
| BLUE | DARK_GREEN | PINK |
| BOOK_BLUE | DARK_RED | RED |
| BOOK_LIGHT_BLUE | GRAY | VIOLET |
| BOOK_RED | GREEN | WHITE |
| CYAN | LIGHT_GRAY | YELLOW |
| DARK_BLUE | MAGENTA | |