

# Kapitel 11.1: Klassen – Einführung

---

Erstelle ein neues Projekt namens **11\_Klassen** und kopiere die Gameloop aus Kapitel 8 in eine neue Datei namens `main.py`. Wenn wir mehrere Formen in unserem Fenster anzeigen wollen, die sich möglicherweise unabhängig voneinander bewegen, dann müssen wir uns für jede dieser Formen einen Punkt im Koordinatensystem, die Größe, die Farbe und evtl. noch weitere Details in Variablen merken. Auf unser Tetris-Spielfeld passen maximal 16 x 20, also 320 Blöcke. Wir bräuchten also 320 x-Variablen, 320 y-Variablen, 320 Farben, usw. Bis ihr das alles eingetippt hättet, hättet ihr längst euer Abitur gemacht! Hierzu muss es eine andere Lösung geben! Die Lösung heißt: **Objekt-Orientierte Programmierung (OOP)** mit Klassen! Hört sich nicht nur geil an, sondern ist es auch! Probiert es selbst aus! In diesem Beispiel ist eine Klasse `Kreis` definiert, von der in jedem 10. Schleifendurchlauf ein neues Kreis-Objekt erzeugt wird, das unabhängig von allen anderen Kreisen hin und her rollt. Achtung: Neben der Datei `main.py` brauchen wir für die Klasse `Kreis` auch noch eine Datei namens `kreis.py`!

## main.py

```
import pygame
from kreis import Kreis

pygame.init()
pygame.display.set_mode((500, 500))
pygame.display.set_caption("Zufällige Kreise")

kreise = []
counter = 0

bRun = True
while bRun:
    pygame.time.delay(100)

    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            bRun = False

    if counter == 0:
        k = Kreis()
        k.set_random_values(500, 500)
        kreise.append(k)

    counter = (counter + 1) % 10

    win = pygame.display.get_surface()
    win.fill((0, 0, 0))

    for k in kreise:
        k.move()
        k.draw(win)
```

```
pygame.display.update()

pygame.quit()
```

## kreis.py

```
class Kreis:
    def __init__(self):
        self.x = 0
        self.y = 0
        self.radius = 10
        self.color = (255, 255, 255)
        self.left = 0
        self.right = 0
        self.speed = 0

        self.move_right = True

    def set_random_values(self, max_x, max_y):
        self.radius = random.randint(10, 50)
        self.speed = random.randint(1, 20)
        self.color = (random.randint(0, 255), random.randint(0, 255),
random.randint(0, 255))
        self.x = random.randint(0, max_x)
        self.left = self.x
        self.y = random.randint(0, max_y)
        self.right = self.x + random.randint(0, max_x - self.x)

    def move(self):
        if self.move_right:
            self.x += self.speed
        else:
            self.x -= self.speed

        if self.x >= self.right:
            self.move_right = False
        elif self.x <= self.left:
            self.move_right = True

    def draw(self, win):
        pygame.draw.circle(win, self.color, (self.x, self.y), self.radius)
```