

Übung 11 Klassen benutzen

REPETITION KLASSEN

Erstelle eine Klasse Dot mit einer x und einer y Koordinate und einer Farbe und einem Konstruktor um diese Attribute zu initialisieren. Erstelle eine Methode mit der ein Kreis an Stelle (x,y) mit der initialisierten Farbe gezeichnet wird.

AUFGABE 1

Erstelle einen Array für 10 Dots. Alle Dot Objekte sollen eine zufällige Position und Farbe erhalten. Zeichne die 10 Dots.

AUFGABE 2

Erstelle 4 Car Objekte. Alle sollen mit dem gleichen Speed initialisiert werden aber jedes soll in einem anderen Viertel der display Window Höhe starten. Benutze einen Array. Lasse alle Fahren. (Car Klasse aus Übung 9, 10 verwenden)

AUFGABE 3

1.

Erstelle eine Klasse Point mit x und y Attributen und den Methoden getX() und getY() die x und y zurückgeben.

Ändere die Dot Klasse so, dass die x und y Koordinate nicht mehr einzelne Attribute sind sondern im Attribut position mit Hilfe von Point vereint werden.

Füge der Klasse Point die Methode addToX(float value) hinzu, welche den x Wert um value erhöht.

Ändere nun auch die Car Klasse so, dass Koordinaten im Attribut position mit Hilfe von Point vereint werden.

2.

Teste was passiert wenn du das Attribut position nicht initialisierst und in der paint() Methode benutzt, also in diesem Fall:

```
1  class Dot(){
2      color clr;
3      Position pos;
4
5      Dot(color clr){
6          this.clr = clr;
7      }
8
9      void paint(){
```

Zusatz: AUFGABE 4

```
class Ball {
    int x, diffx;

    Ball() {
        x= (int) random(1, width);
        diffx= (int) random(1, 5);
    }

    void move(){
        x += diffx;
        if (x<0 || x> width){
            diffx *= -1;
        }
    }

    void paint(){
        ellipse(x, 20, 20, 20);
    }
}
```

Kopiere die Ball Klasse. Erstelle dann ein Ball Objekt und lasse es sich bewegen.
Ändere die Klasse Ball so, dass auch in der y-Achse Bewegung rein kommt.