PROCESSING

EINE EINFÜHRUNG IN DIE INFORMATIK

Created by Michael Kirsch & Beat Rossmy

INHALT

STOFF DER VORLESUNG

PROCESSING BASICS

`key` enthält den Character der zuletzt gedrückten Taste.

```
// 'a' wird gedrückt gehalten
println(key); // -> 'a'
// 'a' wird losgelassen
println(key); // -> 'a'
// 'b' wird gedrückt und losgelassen
println(key); // -> 'b'
```

`keyPressed` gibt an ob gerade mind. eine Taste gedrückt ist.

```
// 'a' wird gedrückt gehalten
println(keyPressed); // -> true
// 'a' wird losgelassen
println(keyPressed); // -> false
// 'b' wird gedrückt und losgelassen
println(keyPressed); // -> true
```

`keyPressed()` wird immer dann aufgerufen wenn eine Taste gedrückt wird.

```
void keyPressed () {
  println(key);
}
```

STRINGS & ARRAYS

Länge eines Strings

```
String s = "abc";
s.length(); // -> 3
```

Index des ersten auftretenden Characters **`c`**

```
s.indexOf('a'); // -> 0
s.indexOf('c'); // -> 2
```

Überprüft ob String **`s`** identisch mit String **`t`** ist.

```
String t = "ab";
s.equals(t); // -> false
t = t + "c";
s.equals(t); // -> true
```

Länge eines Arrays

```
int[] a = new int[] {1,2,3,4,5,6};
a.length; // -> 6
```

KLASSEN

Signalwort + Name + Rumpf-Anfang	`class Kreis {`
Felder	`int x,y,d;`

Konstruktor: Name + Übergabewerte	`public Kreis (intx ,int y, int d) {`
	`...`
	`}`

Methoden	`void plot () {`
	`...`
	`}`

Klassenrumpf-Ende	`}`

OBJEKTE

Objekt deklarieren	`Kreis k1;`
Objekt initialisieren	`k1 = new Kreis(122,321,20);`
Objekt-Methoden Aufruf	`k1.plot();`
Objekt-Feld Zugriff	`println(k1.x);` `k1.x++;`

ÜBUNG

AUFGABE 1

Vier Animationen auf Tastendruck

```
int animationCounter;
void setup () {
  size(600, 400);
  animationCounter = 0;
}
void draw () {
  background(0);
  handleInput();
  drawAnimationA();
}
void handleInput () {
  if (keyPressed) {
    animationCounter++;
  } else {
    animationCounter = 0;
  }
}
void drawAnimationA () {...}
...
```

AUFGABE 1

Vier Animationen auf Tastendruck

```
void drawAnimationA () {
  if (animationCounter>0) {
    // Ein Kreis bewegt sich von links nach rechts
  }
}
void drawAnimationB () {
  if (animationCounter>0) {
    // Ein Kreis schrumpft von einem maximalen Wert zu 0
  }
}
void drawAnimationC () {
  if (animationCounter>0) {
    // Der Hintergrund wechselt zufällig die Farben
  }
}
void drawAnimationD () {
  if (animationCounter>0) {
    // Rechteck wechselt Position bei Tastendruck (gezeichnet wenn gehalten)
  }
}
```

AUFGABE 1-A

Ein Kreis bewegt sich von links nach rechts

```
void drawAnimationA () {
  if (animationCounter>0) {
    // Setze den Counter als x Koordinate der Ellipse
  }
}
```

AUFGABE 1-B

Ein Kreis schrumpft von einem maximalen Wert zu 0

```
void drawAnimationB () {
  if (animationCounter>0) {
    // wenn maximale Größe minus Counter > 0
    // Zeichne Kreis mit diesem Wert als Durchmesser
  }
}
```

AUFGABE 1-C

Der Hintergrund wechselt zufällig die Farben

```
// deklariere Variable für Hintergrundfarbe

void setup () {...}
void draw () {
  // Hintergrund abhängig von Farbvariable
  ...
}

void drawAnimationC () {
  if (animationCounter>0) {
    // wenn Taste gedrückt -> setze Variable für Farbe zufällig
    // ansonsten setze auf Standardfarbe
  }
}
```

AUFGABE 1-D

Neue Position bei Tastendruck (gezeichnet wenn gehalten)

```
// deklariere Variablen zum Speichern der Positionen

void setup () {...}
...

void drawAnimationD () {
  if (animationCounter>0) {
    // erzeuge neue Positionen wenn Taste gerade gedrückt wurde
    // zeichne Rechteck
  }
}
```

AUFGABE 2

Reagiere auf Maus- und Tasteninput.

```
// je ein animationCounter für Mausinput und Tasteninput
void setup () {
  size(600, 400);
  // initialisiere beide animationCounter
}

void draw () {
  background(0);
  // handleMouseInput
  // handleKeyInput
  // drawMouseAnimation
  // drawKeyAnimation
}

void handleMouseInput () {...}
void handleKeyInput () {...}
void drawMouseAnimation () {...}
void drawKeyAnimation () {...}
```

ANIMATIONEN

Versuche dir eigene Animationen auszudenken.

```
void drawAnimationE () {
  if (animationCounter>0) {
    // deine Idee
  }
}
void drawAnimationF () {
  if (animationCounter>0) {
    // deine Idee
  }
}
```