

Graphic Coding

Klausur-Lösungen

9. Februar 2007

Kurs B

Name:

Matrikelnummer:

Hinweise

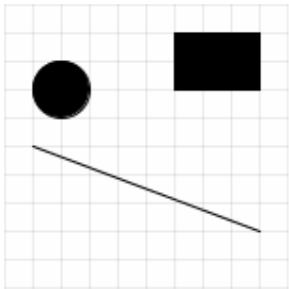
- Es sind keine Hilfsmaterialien erlaubt. (Keine Bücher, Taschenrechner, Handys)
- Sie haben zwei Stunden Zeit.
- Insgesamt sind 100 Punkte zu erreichen. Bei jeder Aufgabe und Teilaufgabe stehen die Punkte anbei. Nutzen Sie diese, um Ihre Zeit einzuteilen.
- Schreiben Sie alle Antworten auf diese Klausurunterlagen. Es sind keine eigenen Zettel erlaubt. Die Rückseiten dürfen für Notizen verwendet werden und werden nicht bewertet. Wenn Sie zusätzliches Papier benötigen, melden Sie sich.
- Keine Panik! Wenn sie eine Aufgabe nicht lösen können, überlegen Sie sich, zur nächsten zu gehen und zum Schluss zu dieser zurück zu kehren.
- Bei etwaigen Fragen melden Sie sich – bitte leise.

Viel Erfolg!

	Punkte	Max
1		12
2		12
3		15
4		18
5		16
6		12
7		5
8		10
Gesamt		100

1. Grundlagen (12 Punkte)

- a) Schreiben Sie Farb- und Zeichenanweisungen, um das folgende Resultat zu erhalten. (Dabei ist das Raster nicht zu programmieren.) (6 Punkte)



```
size(100, 100);  
background(255);  
fill(0);  
rect(60, 10, 30, 20);  
ellipse(20, 30, 20, 20);  
line(10, 50, 90, 80);
```

- b) Beschreiben Sie kurz, wozu die zwei Icons in der Processing Entwicklungsumgebung verwendet werden. (1 Punkt)



Start – Startet das Programm.



Exportieren – Exportiert das Programm fürs Web

- c) Was tun Sie, wenn Sie Auskunft über eine neue, unbekannte Anweisung erhalten möchten. (2 Punkte)

In der Processing-Referenz nachschauen.

- d) Was ist der Unterschied zwischen `setup` und `draw`? (3 Punkte)

`setup()` wird einmal und am Anfang ausgeführt.
`draw()` wird mehrmals wiederholt ausgeführt.

2. Variablen und Datentypen (12 Punkte)

- a) Kreuzen Sie die korrekten der folgenden Deklarationen an. (3 Punkte)

- `float zahl;`
 `char a = 15;`
 `PImage foto;`
 `String txt = "15";`
 `float x == 250.5;`

- b) Welche Ausgabe erzeugt das folgende Programm? (2 Punkte)

```
int a = 0;  
int b = 5;  
a = a + 1;  
b = a * 2 + b;  
b = 10 - b;  
println("b = " + b);  
b = 3
```

2. c) Nennen Sie drei vordefinierte Variablen in Processing und beschreiben Sie kurz, was in diesen gespeichert ist. (3 Punkte)

`width` – Die Weite des Zeichenbereichs.

`keyPressed` – Speichert, ob zur Zeit eine Taste gedrückt ist.

`mouseX` – Die aktuelle x-Position des Mauszeigers.

d) Erläutern Sie den Unterschied zwischen den Werten `15`, `15.0` und `"15"`? (4 Punkte)

Der Datentyp: `int` (17), `float` (17.0), und `String` ("17").

3. Bedingungen (15 Punkte)

a) Wie viele Kreise werden von diesem Programm gezeichnet? (4 Punkte)

```
int a = 4;
int b = 6;

if (a < b) {
    ellipse(0, 0, 10, 10);
}
if (a == b) {
    ellipse(90, 50, 10, 10);
    ellipse(50, 70, 10, 10);
}
else {
    ellipse(80, 50, 10, 10);
}
```

0 1 2 3 4

b) Erweitern Sie den Code so, dass eine farbige Linie vom Mittelpunkt zur aktuellen Mausposition gezeichnet wird, wenn eine Taste auf der Tastatur gedrückt ist. Die RGB-Farbe der Linie soll zufällig ausgewählt werden. (7 Punkte)

```
if ( keyPressed ) {
    stroke(random(255), random(255), random(255));
    line(width/2, height/2, mouseX, mouseY);
}
```

c) Kreuzen Sie die Aussagen an, die richtig sind. (4 Punkte)

- Eine if-Abfrage prüft, ob eine Schleife wiederholt wird.
- Innerhalb eines if-Blocks dürfen mehrere Anweisung stehen.
- Eine Bedingung kann entweder `true`, `false` oder `unkown` sein.
- Es ist nicht möglich zu prüfen, ob zwei Variablen gleich sind.
- Mit Hilfe von `else` werden Anweisungen ausgeführt, wenn die Bedingung nicht erfüllt ist.

4. Dynamik & Bewegung (18 Punkte)

a) Fügen Sie die notwendigen Code-Zeile(n) hinzu, damit sich der Punkt langsam diagonal nach rechts oben bewegt. (4 Punkte)

```
int x = 100;
int y = 100;

void setup() {
    size(200, 200);
}

void draw() {
    point(x, y);

    x = x + 1;

    y = y - 1;
}
```

b) Welche der folgenden Bedingungen prüft, ob ein Element an der Position (x, y) mit einem der Ränder des Zeichenbereichs kollidiert ist? (3 Punkte)

- $y \geq 0$
- $y > \text{height}$
- $y < \text{width}/2$
- $x \leq \text{height}$

c) Welchen der genannten Werte kann die Variable x haben? Kreuzen Sie alle möglichen Werte an. (3 Punkte)

```
float x = 4;
x = x + random(-3, 3);
```

- 2 0 1 4 6.4 7.1

d) Schreiben Sie die draw-Funktion, so dass zufällig positionierte, zufällig große Kreise gezeichnet werden, deren (Graustufen-)Füllfarbe abhängig von der horizontalen Position ist. (8 Punkte)

```
void draw() {

    float s = random(20);

    float x = random(width);
    float y = random(height);

    fill(x);
    ellipse(x, y, s, s);

}
```

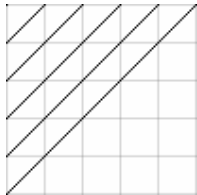
5. Schleifen (16 Punkte)

a) Was ist die Ausgabe des folgenden Programms? (3 Punkte)

```
for (int i = 0; i < 5; i = i + 2) {  
    println(i);  
}
```

0
2
4

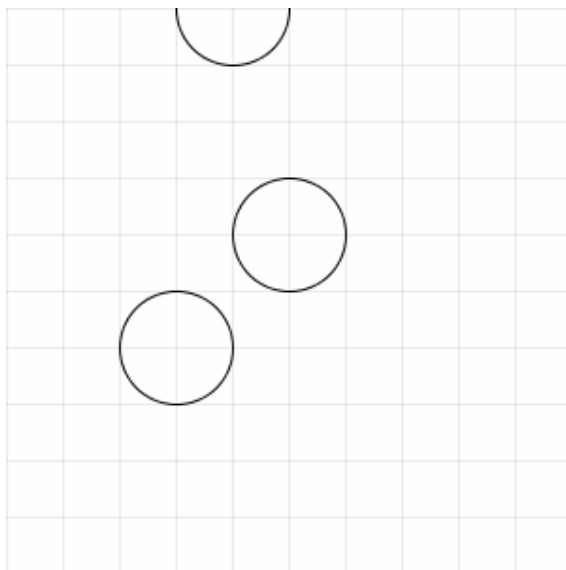
b) Erweitern Sie den Code so, dass folgende 50x50px große Grafik entsteht. (Dabei ist das Raster nicht zu programmieren.) (6 Punkte)



```
int i = 0;  
while (i <= 50) { // oder: i < 60  
    line(i, 0, 0, i); // oder: line(0, i, i, 0);  
    i = i + 10;  
}
```

c) Zeichnen Sie das Ergebnis des Programms innerhalb des 100x100px-Zeichenbereichs. Es ist keine Füllfarbe gesetzt – die Kreise sind also transparent. Bedenken Sie, dass alle int-Variablen mit 0 vordefiniert sind. (7 Punkte)

```
int[] zahlen = new int[6];  
zahlen[3] = 60;  
zahlen[5] = 40;  
int i = 3;  
while (i < 6) {  
    ellipse(i * 10, zahlen[i], 20, 20);  
    i++;  
}
```



6. Funktionen (12 Punkte)

a) Erläutern Sie kurz und knapp die Bedeutung von `void` und nennen Sie je eine grafische Funktion (aus Processing) ohne und eine mit Parametern. (2 + 3 Punkte)

- `void` bedeutet, dass die Funktion keinen Rückgabewert hat.
- ohne Parameter: `noFill()`; mit Parameter: `ellipse(x, y, w, h)`;

b) Definieren Sie eine Funktion `drawBlueCircle`, die einen blauen Kreis an eine bestimmte Position zeichnet. Die Position soll als Fließkomma-Parameter übergeben werden können. Der Kreis soll 15x15 Pixel groß sein. (7 Punkte)

```
void drawBlueCircle(float x, float y) {  
    fill(0, 0, 255);  
    ellipse(x, y, 15, 15);  
}
```

7. Skizzieren Sie das Array zu den beiden Zeitpunkten während des Programmablaufs. Beachten: Alle `int`-Variablen sind mit 0 vordefiniert. (5 Punkte)

```
int[] zahlen = new int[4];  
zahlen[0] = 4;  
zahlen[3] = -1;  


|   |   |   |    |
|---|---|---|----|
| 4 | 0 | 0 | -1 |
|---|---|---|----|

  
zahlen[2] = zahlen[1] + zahlen[0];  
zahlen[3] = 2 * zahlen[3];  


|   |   |   |    |
|---|---|---|----|
| 4 | 0 | 4 | -2 |
|---|---|---|----|


```

8. Bildbearbeitung (10 Punkte)

a) Im folgenden Programm soll ein Bild mit Namen `bild.jpg` geladen und so gezeichnet werden, dass die linke obere Ecke des Bildes an der Position (50, 100) liegt. (6 Punkte)

```
PImage img;  
  
void setup() {  
    size(400, 400);  
  
    img = loadImage("bild.jpg");  
}  
  
void draw() {  
  
    image(img, 50, 100);  
}
```

b) Setzen sie die Füllfarbe, so dass eine gegebene Farbe invertiert wird. Verwenden Sie dabei die drei bereits vorhandenen Variablen `r` (rot), `g` (grün) und `b` (blau). Schreiben Sie genau eine Anweisung.

Tipp: Beim Invertieren werden z.B. weiße Pixel schwarz und blaue Pixel gelb. (4 Punkte)

```
fill(255 - r, 255 - g, 255 - b);
```