

Aufgabenblatt 2

Praktikum am 17.10.2016
Abgabe bis spätestens 23.10.2016

Programmierung
Wintersemester 2016/2017
Prof. Dr. Dirk Eisenbiegler
Hochschule Furtwangen

a) 3 Zahlen

a
b
c

min
max

if (x < 3)

Aufgabe 2.1 - MinMax

- A) Schreiben Sie ein Programm mit dem Klassennamen MinMaxDrei. MinMaxDrei soll von drei Programmparametern das Minimum und das Maximum bestimmen. Zunächst soll das Programm alle drei Programmparameter ausgeben, dann das Minimum und schließlich das Maximum. Testen Sie das Programm mit verschiedenen Eingabewerten. Gehen Sie das Programm mit dem Debugger Schritt für Schritt durch.
- B) Schreiben Sie analog zu Teilaufgabe A) ein Programm mit dem Klassennamen MinMax, das zu einer beliebigen Anzahl von Eingabewerten das Minimum und das Maximum bestimmt.
Hinweis: *args.length* ist die Länge des Arrays *args* - also die Anzahl der Programmparameter.

Aufgabe 2.2 - Palindrom

Schreiben Sie ein Programm, das von einem String prüft, ob er ein Palindrom ist.

Schreiben Sie dazu zunächst eine Methode mit dem Namen *palindrom*. Die Methode *palindrom* hat einen Parameter vom Typ *String* und einen Rückgabewert vom Typ *boolean*, der genau dann den Wert *true* haben soll, wenn der String von links nach rechts gelesen das gleiche Wort ergibt wie von rechts nach links.

Beispiele: "OTTO", "ANNA" und "xYzYx" sind Palindrome, "Katze" ist kein Palindrom.

Die main-Methode soll für den ersten Kommandozeilenparameter entweder den Satz

Der String *xy* ist ein Palindrom.

oder den Satz

Der String *xy* ist kein Palindrom.

ausgeben. Dazu soll in der main-Methode die *palindrom*-Methode verwendet werden.

Hinweise:

- x Ist *x* eine Variable vom Typ *String*, dann ist *x.length()* die Länge des Strings und *x.charAt(i)* der *i*-te Buchstabe (mit $0 \leq i < x.length()$).
- x Den Klassennamen können Sie in dieser Aufgabe und auch in den weiteren Aufgaben frei wählen (soweit nicht explizit angegeben). Es empfiehlt sich als Klassennamen den Aufgabennamen zu verwenden. Allerdings müssen Sie beachten, dass Klassennamen immer mit einem Buchstaben beginnen müssen. Empfehlung: Stellen Sie A oder Aufgabe voran. Außerdem darf der Klassennamen keine Leerzeichen und keine Punkte enthalten. Empfehlung: Sie können stattdessen das Underscore-Zeichen `_` verwenden.

Aufgabenblatt 3

*Praktikum am 24.10.2016
Abgabe bis spätestens 30.10.2016*

*Programmierung
Wintersemester 2016/2017
Prof. Dr. Dirk Eisenbiegler
Hochschule Furtwangen*

Aufgabe 3.1 - Summe, Durchschnitt

- A) Schreiben Sie eine Methode, die zu einem Array von int-Werten die Summe berechnet.
- B) Schreiben Sie eine Methode, die zu einem Array von int-Werten das arithmetische Mittel bestimmt. Verwenden Sie dabei die Methode aus Teilaufgabe A).

Hinweis:

Als Arithmetisches Mittel der Zahlen x_0, x_1, \dots, x_{n-1} bezeichnet man den Wert

$$\frac{x_0 + x_1 + \dots + x_{n-1}}{n} .$$

Aufgabe 3.2 - Muster

Betrachten Sie die folgenden Muster, die aus den Wörtern "Olive" und "Furtwangen" gebildet wurden.

OOOOO
O1111
O1111
O1ivv
Olive

FFFFFFFFF
Fuuuuuuuuu
Furrrrrrrrr
Furtttttttt
Furtwwwww
Furtwaaaaa
Furtwannnn
Furtwanggg
Furtwangee
Furtwangen

Schreiben Sie eine Methode mit dem Namen *muster*, die zu einem beliebigen String ein solches Muster ausgibt.

Aufgabenblatt 4

Praktikum am 31.10.2016

Abgabe bis spätestens 6.11.2016

Programmierung

Wintersemester 2016/2017

Prof. Dr. Dirk Eisenbiegler

Hochschule Furtwangen

Aufgabe 4.1 - Vorkommen von Zeichen zählen

Schreiben Sie eine Methode mit dem Namen `vorkommen`, die einen String zeilenweise Buchstabe für Buchstabe ausgibt und zu jedem Buchstaben angibt, wie oft er im String vorkommt.

Beispiel: Ananasmarmelade

A 1
n 2
a 4
n 2
a 4
s 1
m 2
a 4
r 1
m 2
e 2
l 1
a 4
d 1
e 2

Aufgabe 4.2 - Teilstring suchen | *verbündlich abgeben*

Schreiben Sie eine Methode mit dem Namen *substring*, die in einem String *x* nach einem Teilstring *y* durchsucht. Ist *y* in *x* als Teil einmal oder mehrfach enthalten, so soll die erste Position zurückgegeben an der *y* in *x* beginnt. Die Zählung der Buchstaben beginne beim ersten Buchstaben mit 0. Ist *y* nicht in *x* enthalten, so soll -1 zurückgegeben werden.

Beispiel 1:

x="Hustensaft"

y="Husten"

Ergebnis: 0

Beispiel 2:

x="Hustensaft"

y="saft"

Ergebnis: 6

Beispiel 3:

x="Ananas"

y="na"

Ergebnis: 1

Beispiel 4:

x="Kamel"

y="Pferd"

Ergebnis: -1

abgeben

Aufgabe 4.3 - Nullstellensuche durch Intervallschachtelung

Schreiben Sie zwei Methoden mit den Namen *f* und *nullstelle*.

Die Methode *f* soll die Funktion $f(x) = e^x + x^2 - 4$ realisieren. Der Parameter und der Rückgabewert der Funktion *f* sollen beide vom Typ *double* sein.

Die Methode *nullstelle* soll in einem Intervall $[x, y]$ per Intervallschachtelung nach einer Nullstelle suchen. Das Iterationsverfahren soll erst dann abbrechen, wenn ein Wert *m* gefunden wird, sodass $|f(m)| < z$ gilt. Ist ein solcher Wert *m* gefunden, so soll die Methode diesen Wert zurückgeben. Die Werte *x*, *y* und *z* sind Parameter der Methode *nullstelle*. Alle Parameter und der Rückgabewert von *nullstelle* seien vom Typ *double*. Implementieren Sie die Methode *nullstelle* mit Hilfe von Rekursion.

Testen Sie die Methode *nullstelle* mit den Werten $x = -1.0$, $y = 20.0$, $z = 0.0001$ sowie mit den Werten $x = -100.0$, $y = 0.0$, $z = 0.0001$.

Hinweise:

x Die Java-Funktion `Math.exp(x)` berechnet e^x .

x Die Java-Funktion `Math.abs(x)` berechnet $|x|$.

Es darf angenommen werden, dass von den beiden Werten $f(x)$ und $f(y)$ einer der beiden Werte größer als 0 ist und der andere kleiner als 0.

Aufgabenblatt 5

Praktikum am 7.11.2016

Abgabe bis spätestens 13.11.2016

Programmierung

Wintersemester 2016/2017

Prof. Dr. Dirk Eisenbiegler

Hochschule Furtwangen

Aufgabe 5.1 - Rekursion (Fibonacci)

Rekursion \Rightarrow selber aufrufen

```
public class Fibonacci {  
    public static int fibonacci (int n) {  
        if (n == 0)  
            return 1;  
        if (n == 1)  
            return 1;  
        return fibonacci(n-1) + fibonacci(n-2);  
    }  
}
```

$\begin{array}{cc} 4-1 & 4-2 \\ 3 & 2 \end{array}$

$n=4$

- A) Ohne das Programm zu starten: Bestimmen Sie die Werte von `Fibonacci.fibonacci` für $n = 0, 1, 2, 3, 4, 5, 7$.
- B) Wie verhält sich die Methode `Fibonacci.fibonacci`, wenn sie mit dem Parameter `-1` gestartet wird? *Stack overflow*
- C) Implementieren Sie `Fibonacci.fibonacci` neu. Die neue Implementierung soll ohne Rekursion auskommen. Für Zahlen kleiner als 0 soll der Rückgabewert 0 sein - für Zahlen größer oder gleich 0 soll der Rückgabewert wie in der bisherigen Implementierung sein.

Aufgabe 5.2 - Rekursion (Fakultät)

In dieser Aufgabe soll die Fakultät $n!$ einer natürlichen Zahl n auf zwei unterschiedliche Arten berechnet werden. Unter der Fakultät einer natürlichen Zahl n versteht man das Produkt aller Zahlen von 1 bis n .

- A) Schreiben Sie eine Methode, die die Fakultät berechnet. Der einzige Parameter ist eine int-Zahl n , der Rückgabewert ist vom Typ int. Berechnen Sie das Ergebnis, indem Sie in einer for-Schleife die Zahlen von 1 bis n in einer Variable aufmultiplizieren.
- B) Schreiben Sie eine Methode mit der gleichen Funktion wie in Teilaufgabe A). Berechnen Sie die Fakultät diesmal jedoch ohne for-Schleife, sondern mit Rekursion. Verwenden Sie die folgende Rekursionsgleichung:

$$n! = \begin{cases} 1 & \text{falls } n=1 \\ (n-1)! \cdot n & \text{falls } n>1 \end{cases}$$

Aufgabenblatt 6

Praktikum am 14.11.2016

Abgabe bis spätestens 20.11.2016

Programmierung

Wintersemester 2016/2017

Prof. Dr. Dirk Eisenbiegler

Hochschule Furtwangen

Aufgabe 6.1 - Library einbinden

A) Library in das Eclipse-Projekt kopieren

Im Verzeichnis /home/hfu/picture befindet sich die Library mit dem Dateinamen picture.jar. Fügen Sie diese Datei zu dem Eclipse-Projekt hinzu. Gehen Sie dazu in das Dateien-Werkzeug (Icon auf der linken Leiste im Desktop), wählen Sie die Datei aus und kopieren Sie diese: Datei anklicken, rechte Maustaste, kopieren. Gehen Sie dann zu Eclipse, wählen Sie das aktuelle Projekt (Name: Programmierung) und führen Sie dort eine Einfügeoperation durch: rechte Maustaste, „einfügen“.

B) Library zum Build-Path hinzufügen.

In dem Eclipse-Projekt „Programmierung“ ist jetzt die Datei picture.jar zu sehen. Wählen Sie diese Datei mit der Maus aus, drücken Sie die rechte Maustaste und wählen Sie im Menü den Punkt „Build-Path“ und dort „Add To Build Path“. Ab sofort können Sie in dem Projekt die in der Library enthaltenen Klassen verwenden.

Aufgabe 6.2 - Bildverarbeitung

In dieser Aufgabe sollen die beiden Methoden `Picture.load` und `Picture.show` verwendet werden. Die Klasse `Picture` ist Bestandteil von `picture.jar`.

Der Methodenaufruf

```
Picture.load("/home/hfu/picture/MyPicture.jpg")
```

lädt eine Bilddatei und gibt als Rückgabewert ein zweidimensionales Array `p` vom Typ `int[][]` zurück, in dem die Pixel des Bildes abgespeichert sind. Mit dem Befehl `Picture.show(p)` kann das Bild angezeigt werden.

```
int p[][] = Picture.load("/home/hfu/picture/MyPicture.jpg");  
Picture.show(p);
```

In der Array-Variablen `p[x][y]` befindet sich die Farbinformation für den Punkt mit den Koordinaten (x,y) . Die Punkte sind in dem zweidimensionalen Array wie folgt angeordnet:

↙ ↘

<code>p[0][0]</code>	<code>p[1][0]</code>	<code>p[2][0]</code>	<code>p[3][0]</code>	<code>p[4][0]</code>	<code>p[5][0]</code>
<code>p[0][1]</code>	<code>p[1][1]</code>	<code>p[2][1]</code>	<code>p[3][1]</code>	<code>p[4][1]</code>	<code>p[5][1]</code>
<code>p[0][2]</code>	<code>p[1][2]</code>	<code>p[2][2]</code>	<code>p[3][2]</code>	<code>p[4][2]</code>	<code>p[5][2]</code>
<code>p[0][3]</code>	<code>p[1][3]</code>	<code>p[2][3]</code>	<code>p[3][3]</code>	<code>p[4][3]</code>	<code>p[5][3]</code>

Farbwerte
int

Die Breite des Bildes ist `p.length`, die Höhe ist `p[0].length`. Damit liegt x im Bereich von 0 bis `p.length-1` und y im Bereich 0 bis `p[0].length-1`.

In den nachfolgenden Teilaufgaben sollen die Bilder durch Programme verändert werden, indem die Array-Werte verändert werden. Das Programm soll immer folgenden Aufbau haben: das Bild wird eingelesen, dann verändert, dann angezeigt.

$$x = p.length - 1$$
$$y = p[0].length - 1$$


```
int p[1][1] = Picture.load("/root/picture/MyPicture.jpg");
// hier ist der Code zur Änderung des Bildes einzufügen
Picture.show(p);
```

Zur Illustration der Transformationen wird das folgende Beispielbild verwendet:



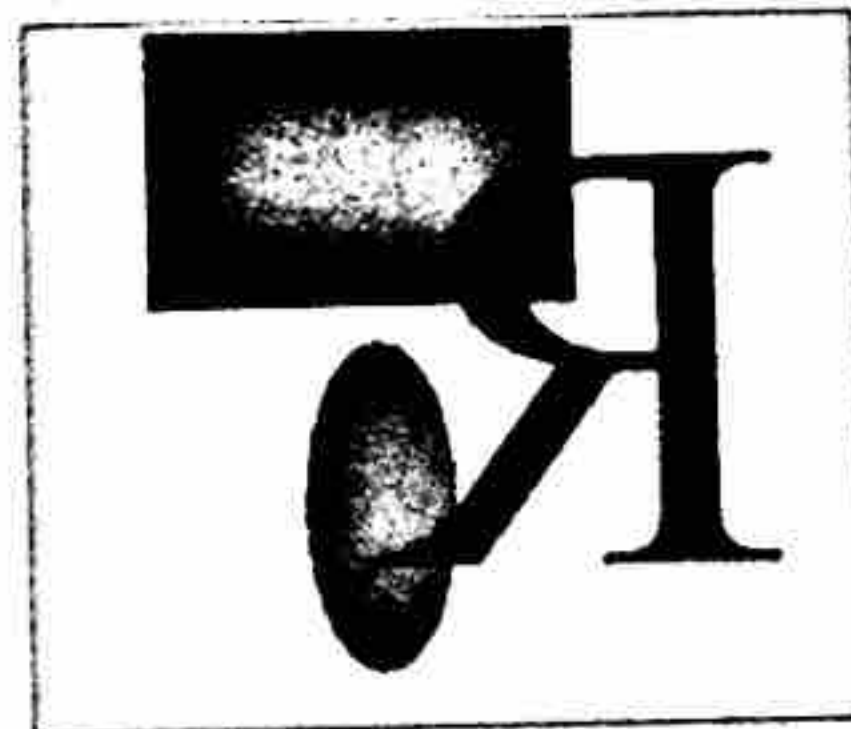
- A) Fügen Sie ein schwarzes Rechteck ein. Die Koordinaten der linken oberen Ecke seien (30,70), die der Ecke rechts unten (250,160). Die Farbe Schwarz wird durch den Wert 0 repräsentiert.

30, 70
250, 160



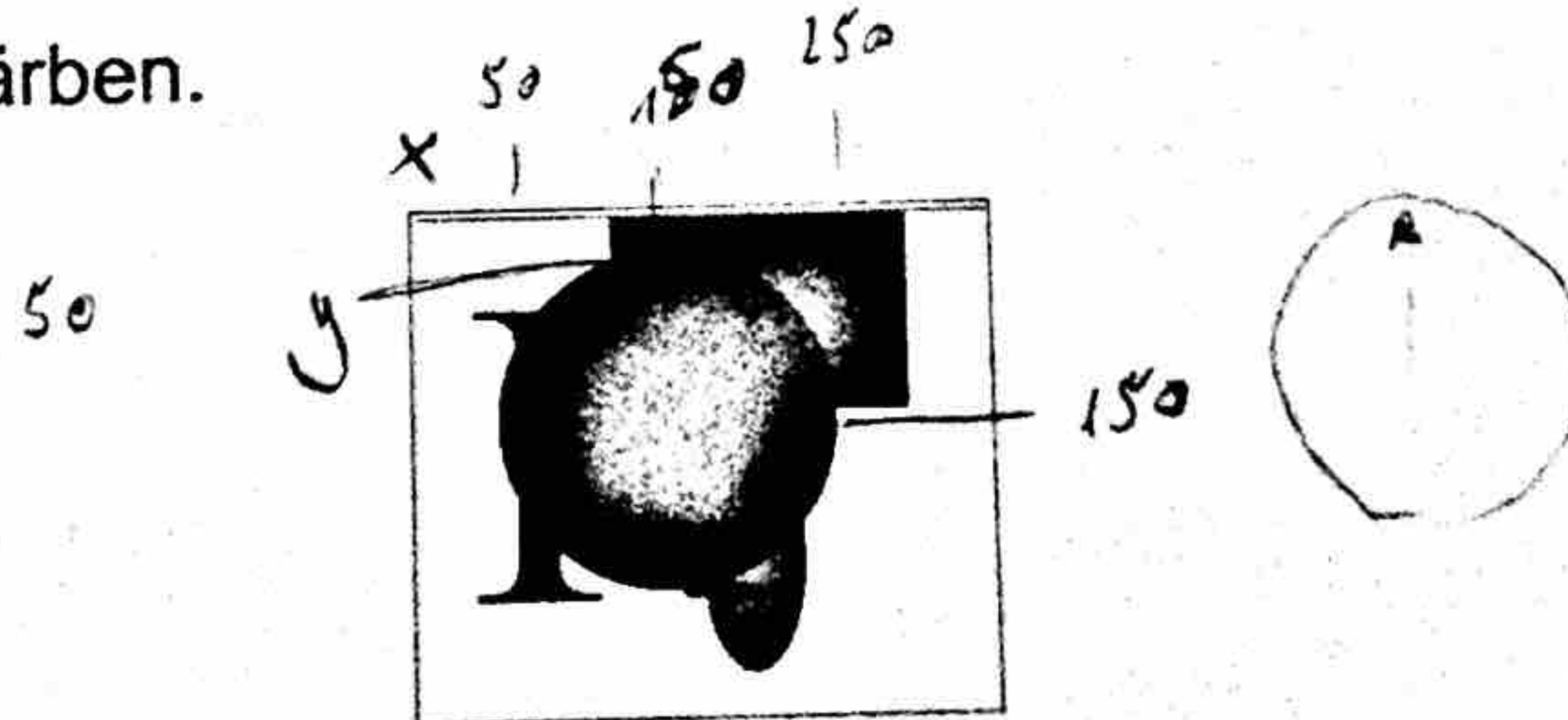
Rechteck
- 1 - x

- B) Spiegeln Sie das Bild horizontal.



- C) Färben Sie alle Punkte in einer Kreisfläche schwarz. Mittelpunkt der Kreisfläche ist (150,150), der Radius ist 100. Hinweis: Bestimmen Sie zu jedem Punkt den Abstand zum Kreismittelpunkt und entscheiden Sie dann, ob Sie ihn schwarz färben.

$p_1 [150] [150]$

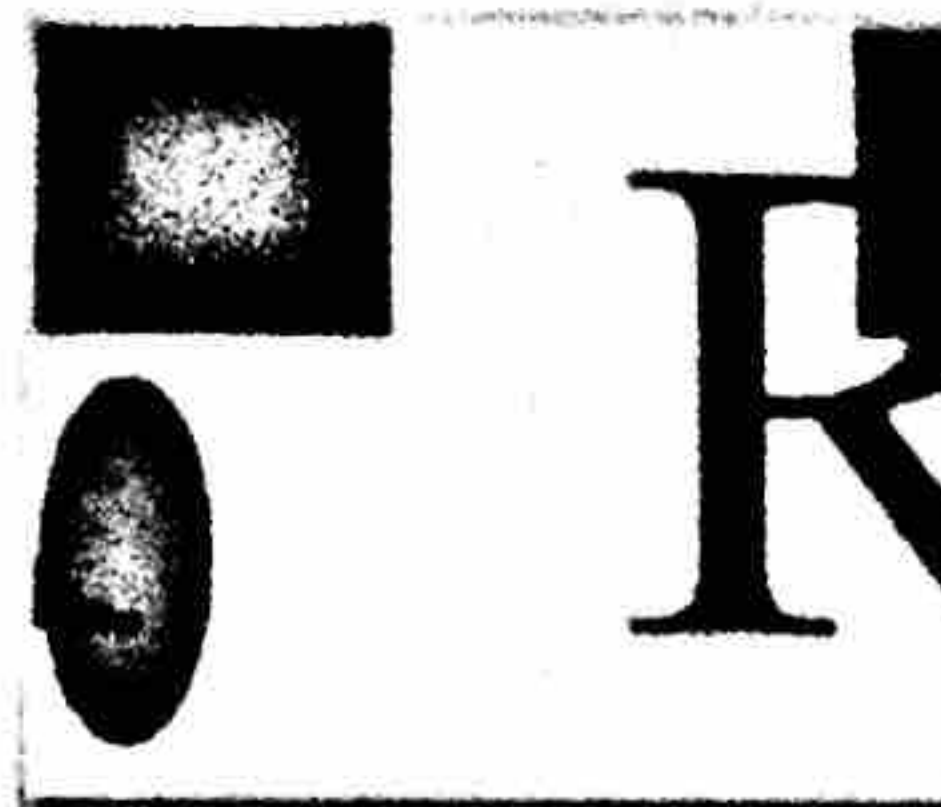


- D) Gegeben der Kreis mit Mittelpunkt (150,150) und dem Radius 100. Spiegeln Sie innerhalb des Kreise alle Punkte an einer horizontalen Achse, die durch den Mittelpunkt geht.



- E) Verschieben Sie das Bild um 180 Punkte nach rechts. Lassen Sie das Bild dabei "rotieren", sodass die nach rechts hinausgeschobenen Punkte von links in das Bild geschoben werden.**

Hinweis: Führen Sie die Rotation Zeile für Zeile aus. Implementieren Sie zunächst eine Rotation um einen Punkt.



- F) Scheren Sie das Bild um 45 Grad.**

