Praktikum Computergrafik, Blatt 3

** Aufgabe 1 (Der Cohen-Sutherland-Algorithmus)

In Moodle ☜ finden Sie die Datei `cohensutherland.zip` mit den Rahmenprogrammen:

`CohenSutherlandPanel.java`: `JPanel`-Klasse mit `main`-Routine zur Darstellung , muss nicht editiert werden, Sie können hier aber die dargestellte(n) Linie(n) zum Testen ändern (s.u.).

`CohenSutherland.java`: Das ist die Klasse, in der Sie den Cohen-Sutherland-Algorithmus implementieren müssen.

`CohenSutherlandTest.java`: JUnit4-Test, muss nicht editiert werden. Identisch mit dem vom APA-Server ausgeführten Test.

`Area.java`: Enthält Konstanten, mit denen die 9 Areale des Cohen-Sutherland-Algorithmus charakterisiert werden können.

a) Implementieren Sie in der Klasse `CohenSutherland` zunächst die Methode

```
int outputCode(int x, int y)
```

die für einen gegebenen Punkt `(x,y)` den Outputcode für Cohen-Sutherland berechnet. Der erwartete Rückgabewert wird im JavaDoc-Kommentar der Methode beschrieben. Zum Setzen der Bits nutzen Sie die Konstanten aus der Klasse `Area`.

Wer „*magic numbers*" benutzt, disqualifiziert sich.

b) Implementieren Sie dann in der Klasse `CohenSutherland` die rekursive Methode

```
void clipLine(int xA, int yA, int xE, int yE)
```

die eine durch Anfangspunkt (x_A, y_A) und Endpunkt (x_E, y_E) gegebene Linie clippt und zeichnet.

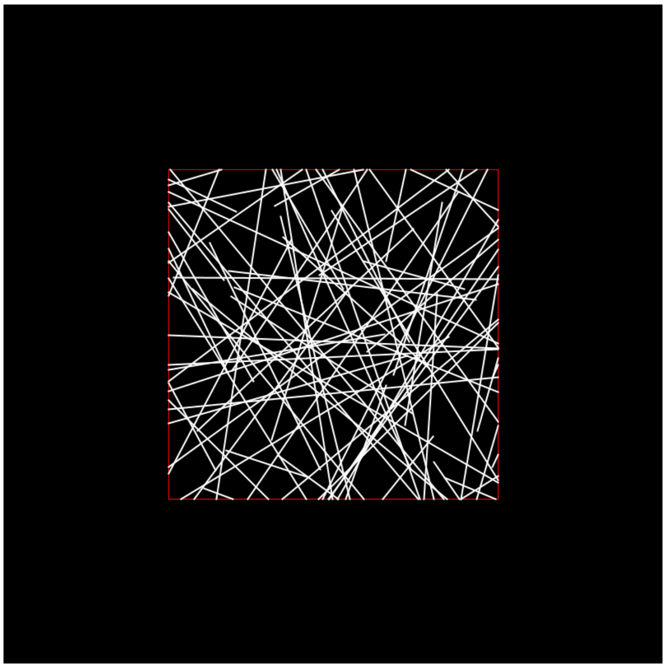
Verwenden Sie die Formeln für die Schnittpunktberechnung aus dem Skript. Da alle Werte ganzzahlig sind, ist es das Ergebnis auch automatisch. Beachten Sie bezüglich der Reihenfolge von Multiplikation und Division die Eigenheiten der Ganzzahldivision.

c) Wenn alles gut geht, dann sollte die Ausführung von `main` in `CohenSutherlandPanel.java` das rote Clipping-Fenster zeigen und darin eine weiße Linie, die komplett gezeichnet wird. Dies erfolgt durch die Code-Zeile

```
cohenSutherland.clipLine(300, 300, 500, 500);
```

Modifizieren Sie Anfangs- und Endpunkt und testen Sie so, ob ihr Clipping funktioniert.

Unterhalb der Zeile befindet sich ein Code-Schnippsel, der 100 Zufallslinien clippen und zeichnen soll. Kontrollieren Sie, ob Sie dieses Ergebnis erhalten:



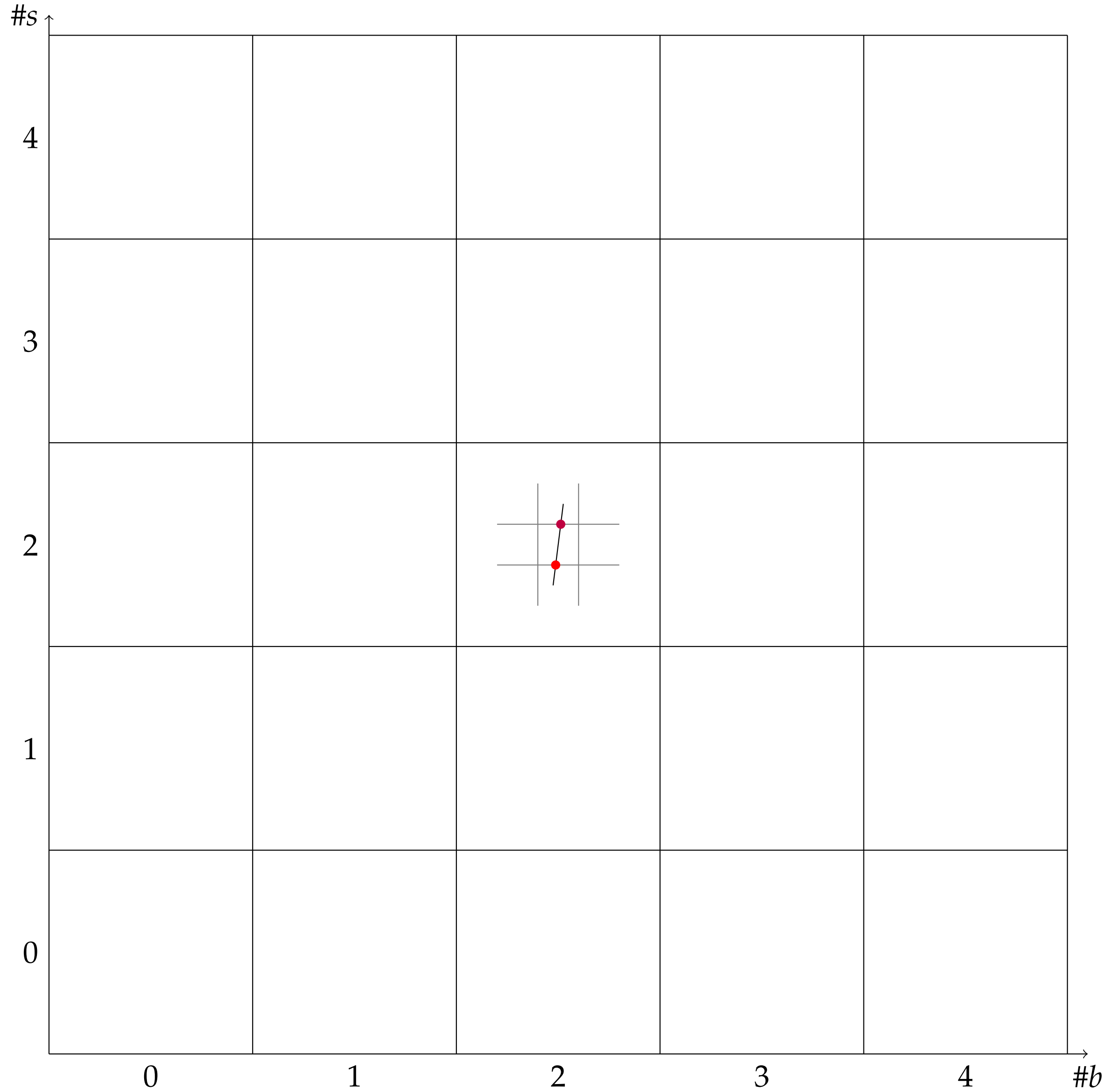
Fragen zur Übung (prüfungsrelevant)

1. Wir legen die in der Vorlesung willkürlich festgelegte Reihenfolge der Bits und ihrer Bedeutung in den Output-Codes von Cohen-Sutherland zugrunde ($y > y_{\max}$, $y < y_{\min}$, $x > x_{\max}$, $x < x_{\min}$).

   (a) Sei „#*b*" die Anzahl der gesetzten Bits in der Oder-Verknüpfung der Output-Codes von Anfangs- und Endpunkt, weiter sei „#*s*" die Anzahl der von Cohen-Sutherland vorgenommenen Schnittpunktberechnungen mit dem Clipping-Rechteck.

   Tragen Sie, falls der Fall auftreten kann, eine Skizze einer derartigen Konstellation ein. Codieren Sie farbig, in welcher Reihenfolge die Schnittpunkte berechnet werden.

   Ein beispielhafter Eintrag ist bereits vorgegeben.



● 1. . . .   ● 2. . . .   ● 3. . . .   ● 4. berechneter Schnittpunkt

(b) Falls Sie keine Konstellation für ein Paar (#*b*, #*s*) finden konnten, geben Sie eine Begründung dafür an, warum der Fall nicht auftreten kann.

2. Inverse affine Transformationen

   Eine zweidimensionale affine Transformation in homogenen Koordinaten sei durch eine 3×3-Matrix A beschrieben. Zeigen Sie in allgemeiner Darstellung für Translation, Skalierung, Scherung und Rotation:

   > Die A rückgängig machende affine Transformation ist zugleich auch die zu A inverse Matrix A^{-1}.

   Z.B. wird die Translation um den Vektor (2, 3) nachvollziehbar durch die Translation um (-2, -3) rückgängig gemacht. Der Satz behauptet also, dass zu $A = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}$ die inverse Matrix

   $A = \begin{pmatrix} 1 & 0 & -2 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{pmatrix}$ sei.

   Zur Erinnerung: Das Produkt einer Matrix A mit ihrer inversen Matrix A^{-1} ergibt die Einheitsmatrix E (Diagonalmatrix mit nur 1en in der Diagonale):

   $$A \cdot A^{-1} = E$$