

**Polygone, Clipping, Polygonoffsetting**

*Sergej Petuhov*  
&  
*Paul Frühauf*

**Proseminar: 3D-Druck-Verfahren**



# Gliederung

## I. Polygone

- 1). Definition
- 2). Konkav, konvex und überschlagen

## II. Clipping

- 1). Was ist Clipping
- 2). Clipping von Strecken
- 3). Polygonclipping
- 4). Nutzen von Clipping

## III. Polygonoffsetting

- 1). Polygonoffsetting
- 2). Umlaufzahlen
- 3). Inneres/- Äußeres Offset



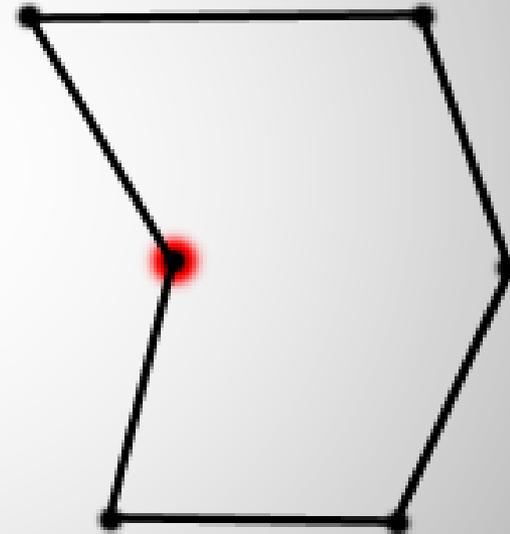
# Definition

Ein **Polygon** ist eine Figur, die durch ein Tupel von verschiedenen Punkten **definiert** ist. Die Punkte heißen die Eckpunkte oder kurz Ecken des **Polygons**, ein **Polygon** mit  $n$  Ecken heißt auch ...-Eck. (Poly = Viel; gon = eck)

# Konkav



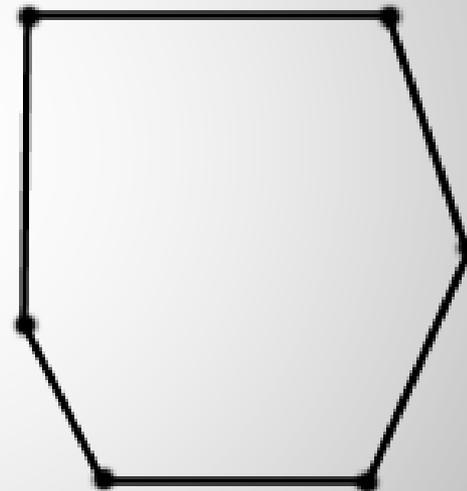
- Nach innen gewölbt
- Innenwinkel über  $180^\circ$
- Mind. 2 Seiten nach innen gewölbt





# Konvex

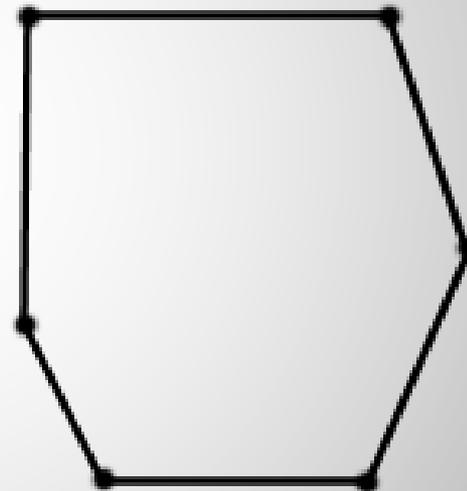
- Nach außen gewölbt
- Innenwinkel unter  $180^\circ$
- Mind. ein Innenwinkel über  $180^\circ$  != konvex





# Konvex

- Nach außen gewölbt
- Innenwinkel unter  $180^\circ$
- Mind. ein Innenwinkel über  $180^\circ$  != konvex

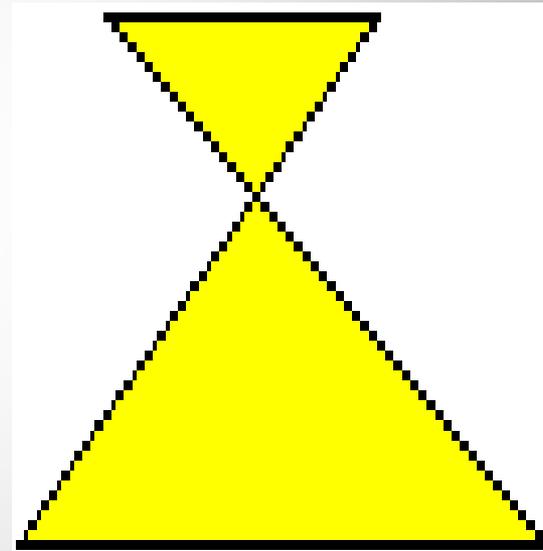


Ist der Bauch konkav, war das Mädchen brav. Ist der Bauch konvex, hatte das Mädchen Sex.

# Überschlagen

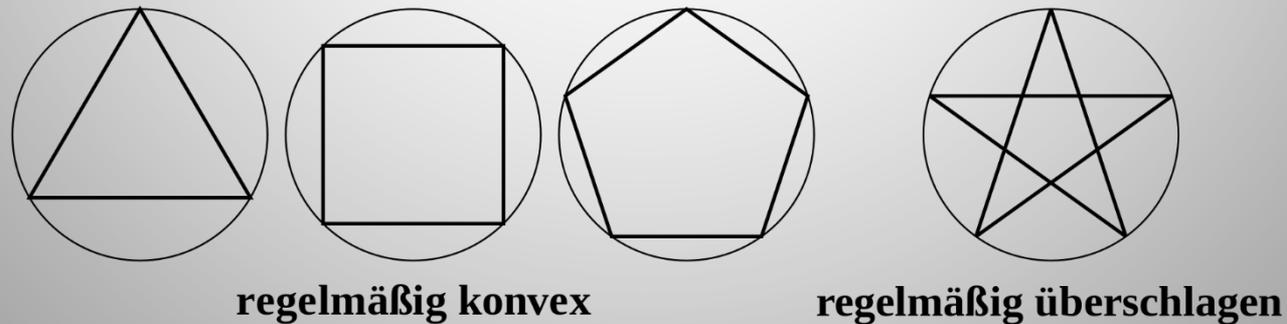
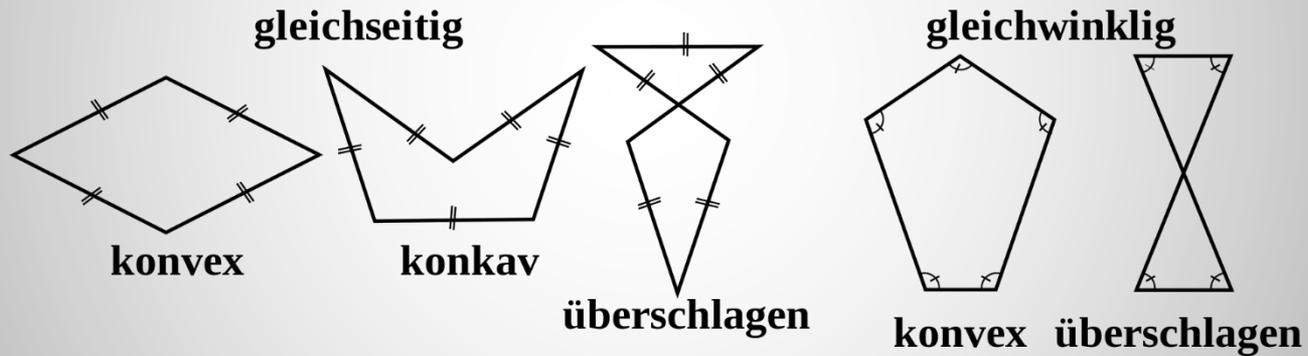
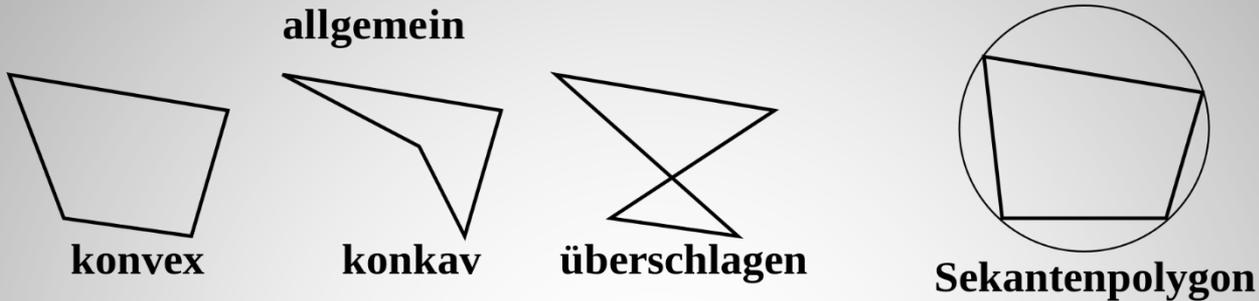


- Kanten berühren sich nicht nur in den Eckpunkten
- Das Gegenteil von einfachen Polygonen





# Zusammenfassung





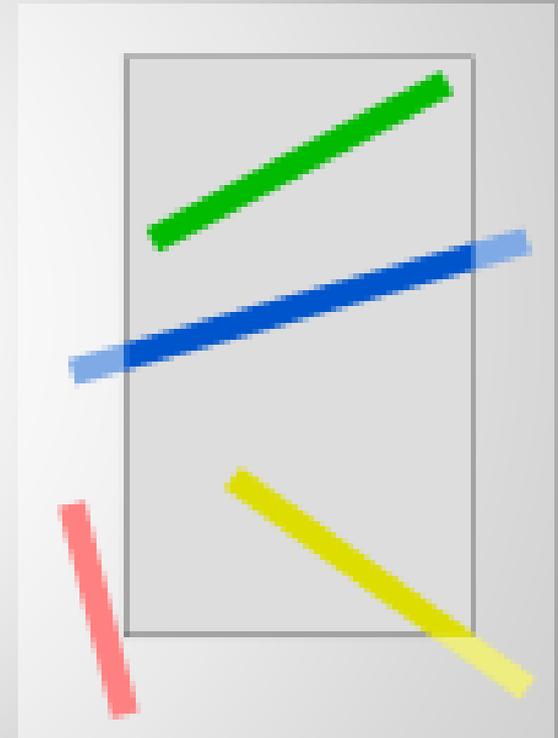
# Was ist Clipping

„**Clipping**“ bedeutet das Abschneiden eines **Grundobjektes** anhand eines **Bildschirmfensters**. Ein Fenster kann dabei ein beliebiges Polygon sein. Das sog. **Clip Polygon**.



# Clipping von Strecken

- P und Q sind beide Endpunkte einer Strecke
- P und Q liegen im Rechteckfenster (grün)
- P und Q liegen außerhalb des Rechteckfensters. (Das Fenster durchquerend) (blau)
- P oder Q liegen im Rechteckfenster (gelb)
- P und Q liegen außerhalb des Rechteckfensters (rot)

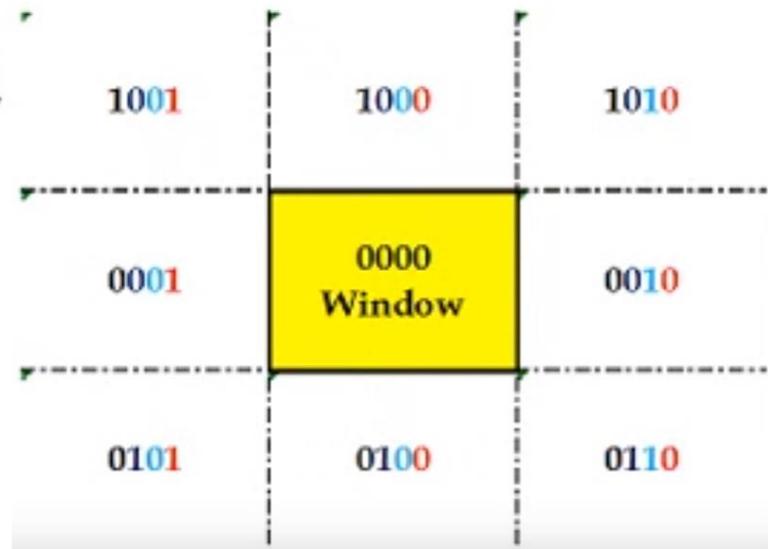




# Clipping von Strecken Algorithmus

## Cohen-Sutherland Algorithmus

$b_1 = 1$  if the point is in the left side of the window  
 $b_2 = 1$  if the point is in the Right side of the window  
 $b_3 = 1$  if the point is in the Bottom side of the window  
 $b_4 = 1$  if the point is in the Top side of the window





# Clipping von Strecken Algorithmus

**Schritt 1:** Zwei Endpunkte bestimmen  $(x_1, y_1)$  und  $(x_2, y_2)$

**Schritt 2:** Outcode von den Endpunkten bestimmen

- Outcode der Endpunkte = 0000 => Linie innerhalb
- AND Operation beider Outcodes != 0000 => Linie außerhalb

**Schritt 3:**

- $x_{\text{Min}}/x_{\text{Max}}$  überschritten =>  $y = y_1 + m (x - x_1)$
- $y_{\text{Min}}/y_{\text{Max}}$  überschritten =>  $x = x_1 + (y - y_1) / m$

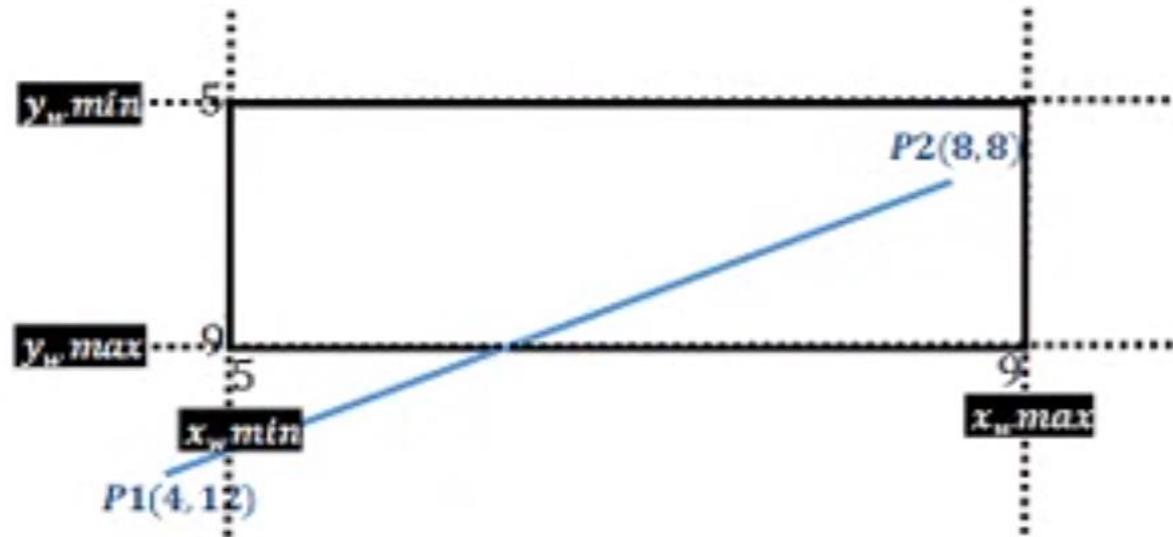
**Schritt 4:**

*Wenn der neu berechnete Punkt  $(x, y)$  die folgenden Bedingungen nicht erfüllt, wiederhole Schritt 3.*

- $x_{\text{Min}} \leq x \leq x_{\text{Max}}$
- $y_{\text{Min}} \leq y \leq y_{\text{Max}}$



# Clipping von Strecken Algorithmus

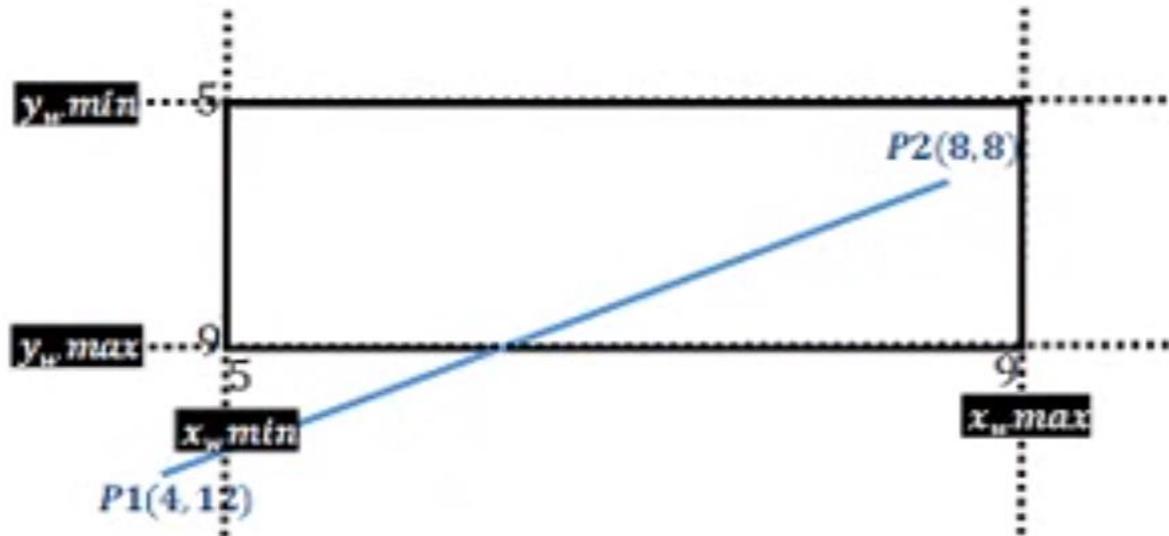


$$x_1 = 4 \quad y_1 = 12 \quad x_2 = 8 \quad y_2 = 8$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{8 - 12}{8 - 4} = -1$$



# Clipping von Strecken Algorithmus



Find  $y$

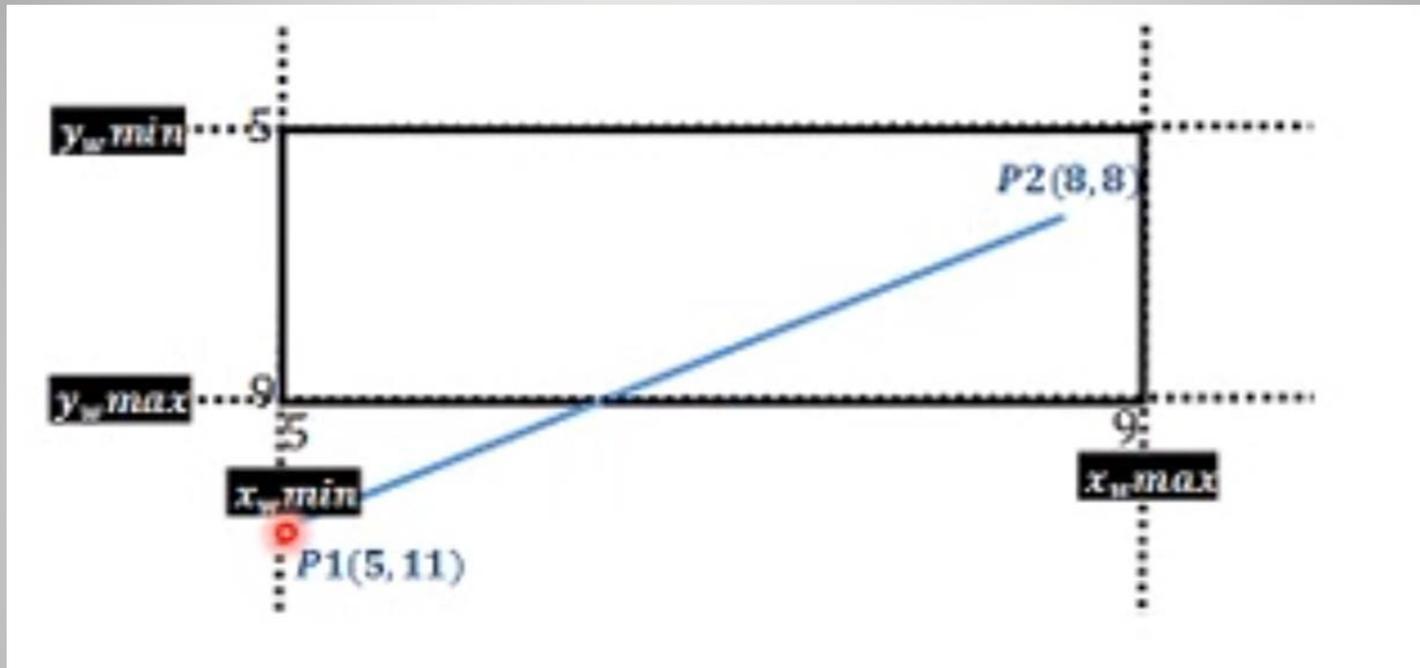
$$y = y_1 + m(x - x_1)$$

Here  $x = x_w.min = 5$

$$y = y_1 + m(x - x_1) = 12 + (-1)(5 - 4) = 12 - 1 = 11$$



# Clipping von Strecken Algorithmus



$$x_w \min \leq x \leq x_w \max$$

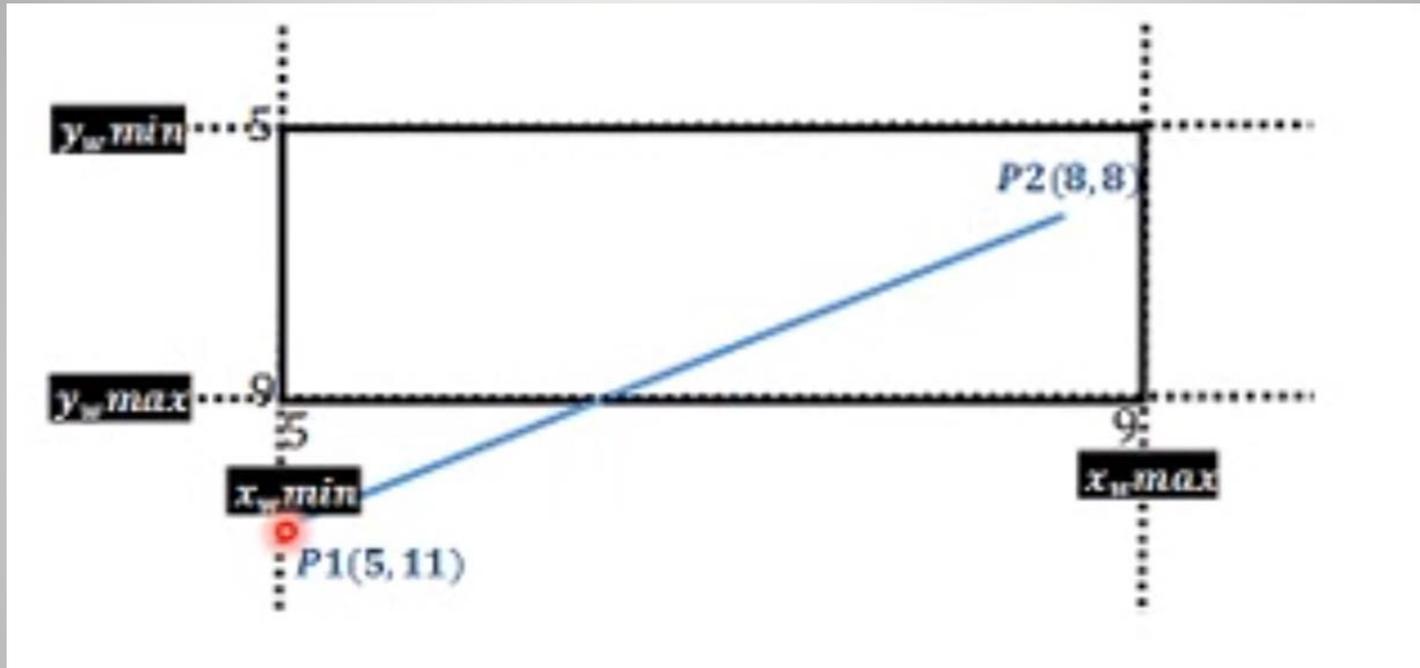
$$y_w \min \leq y \leq y_w \max$$

$$x_w \min \leq 5 \leq x_w \max$$

$$y_w \min \leq 11 \leq y_w \max$$



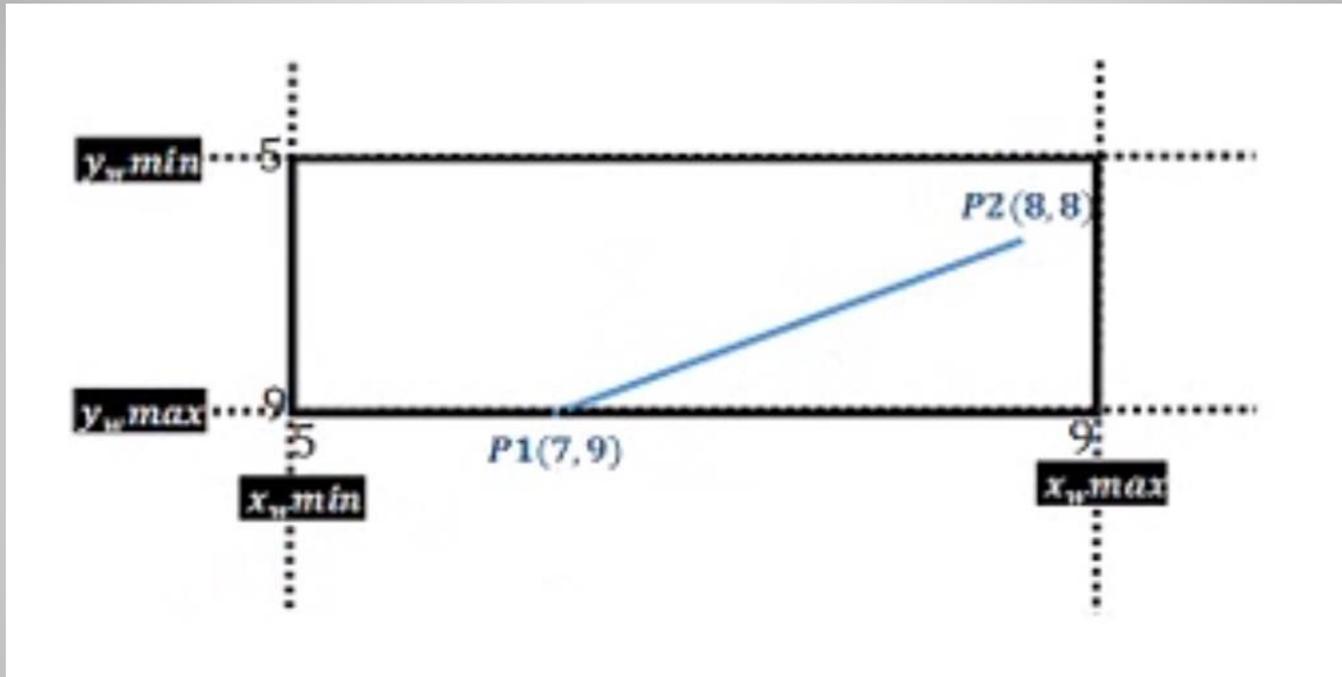
# Clipping von Strecken Algorithmus



$$\begin{aligned}x &= x_1 + \frac{(y - y_1)}{m} \\ &= 5 + \frac{(9 - 11)}{-1} = 5 + 2 = 7\end{aligned}$$



# Clipping von Strecken Algorithmus

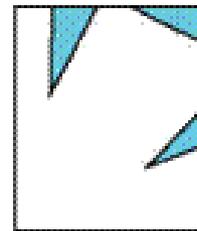
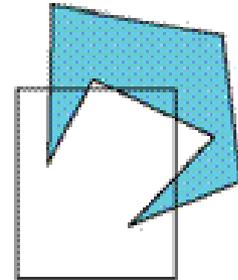


$$\begin{aligned}x_wmin &\leq 7 \leq x_wmax \\y_wmin &\leq 9 \leq y_wmax\end{aligned}$$



# Polygonclipping

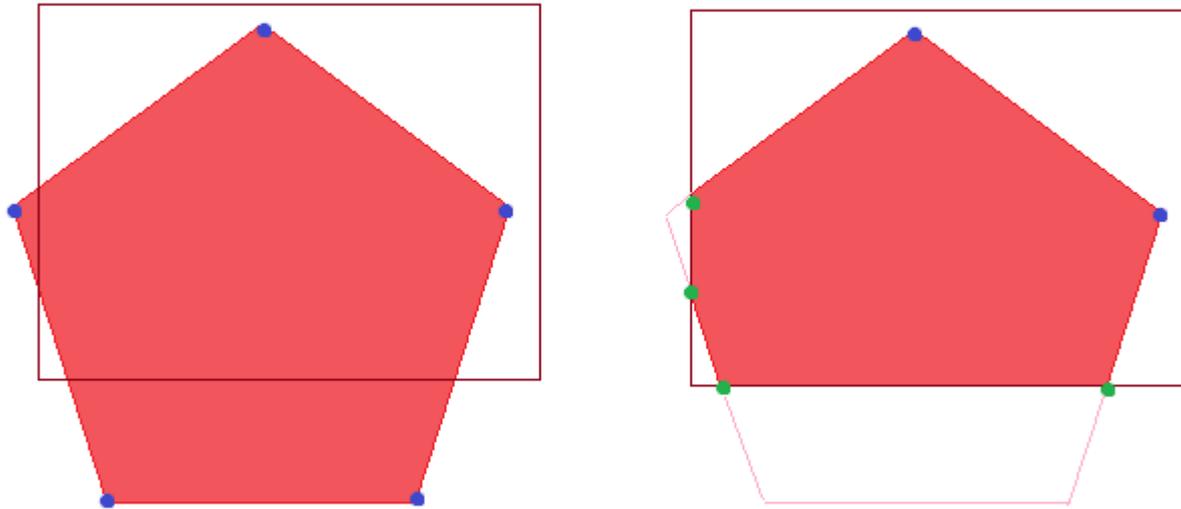
- Kanten des Polygons müssen vollständig, teilweise oder gar nicht übernommen werden
- Neue Kanten müssen aufgenommen werden
- Aus einem Polygon können beim Clipping mehrere Polygone entstehen



# Polygonclipping



## Sutherland-Hodgman Algorithmus





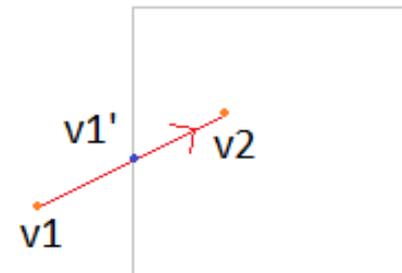
# Polygonclipping

1)



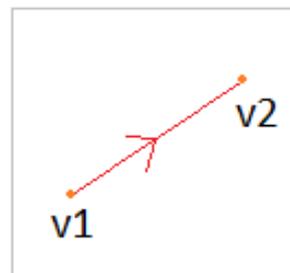
v1 -outside  
v2 -outside

2)



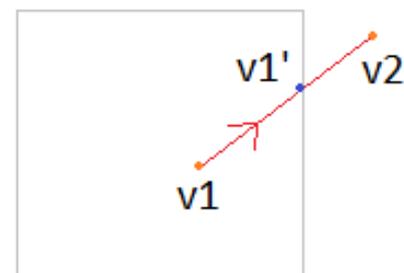
v1 -outside  
v2 -inside

3)



v1 -inside  
v2 -inside

4)

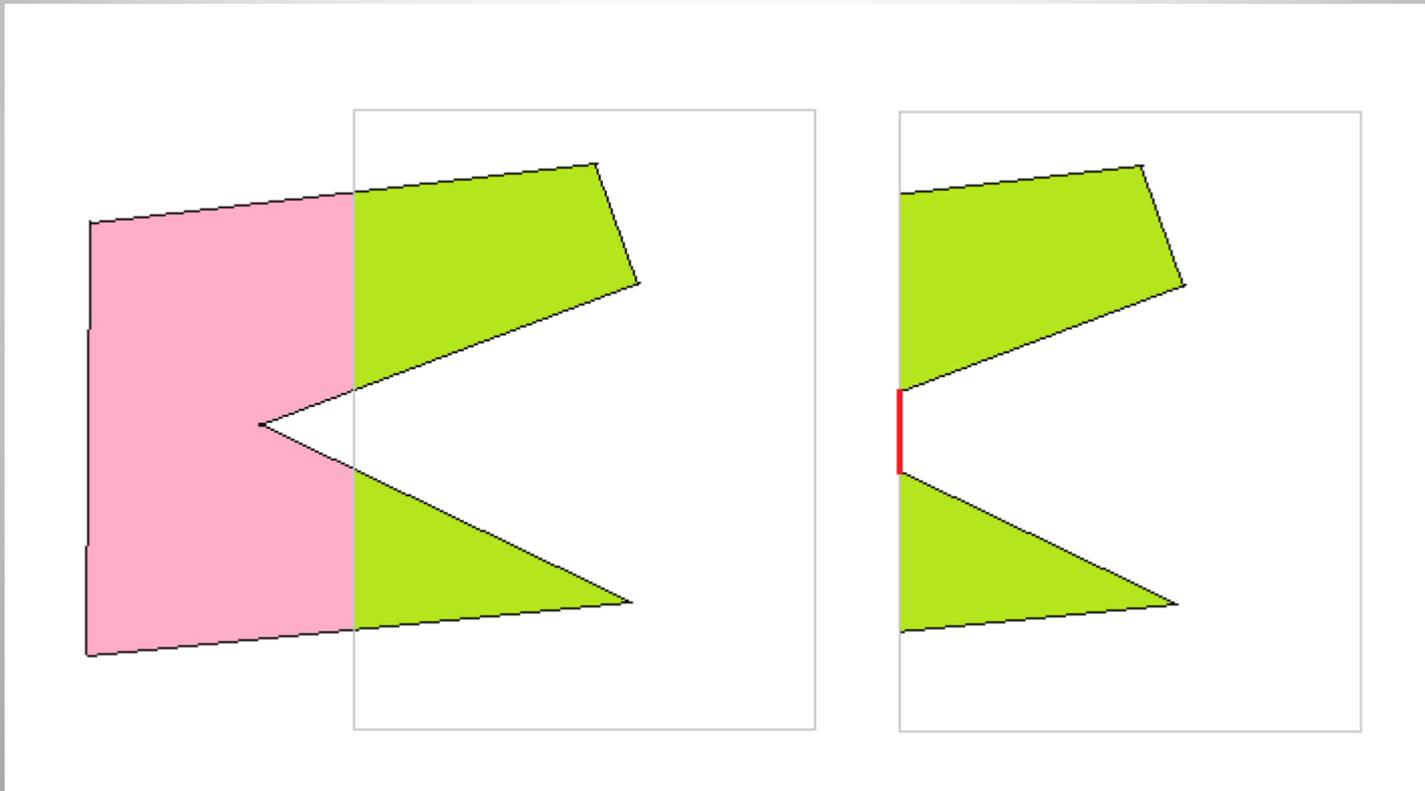


v1 -inside  
v2 -outside

# Polygonclipping

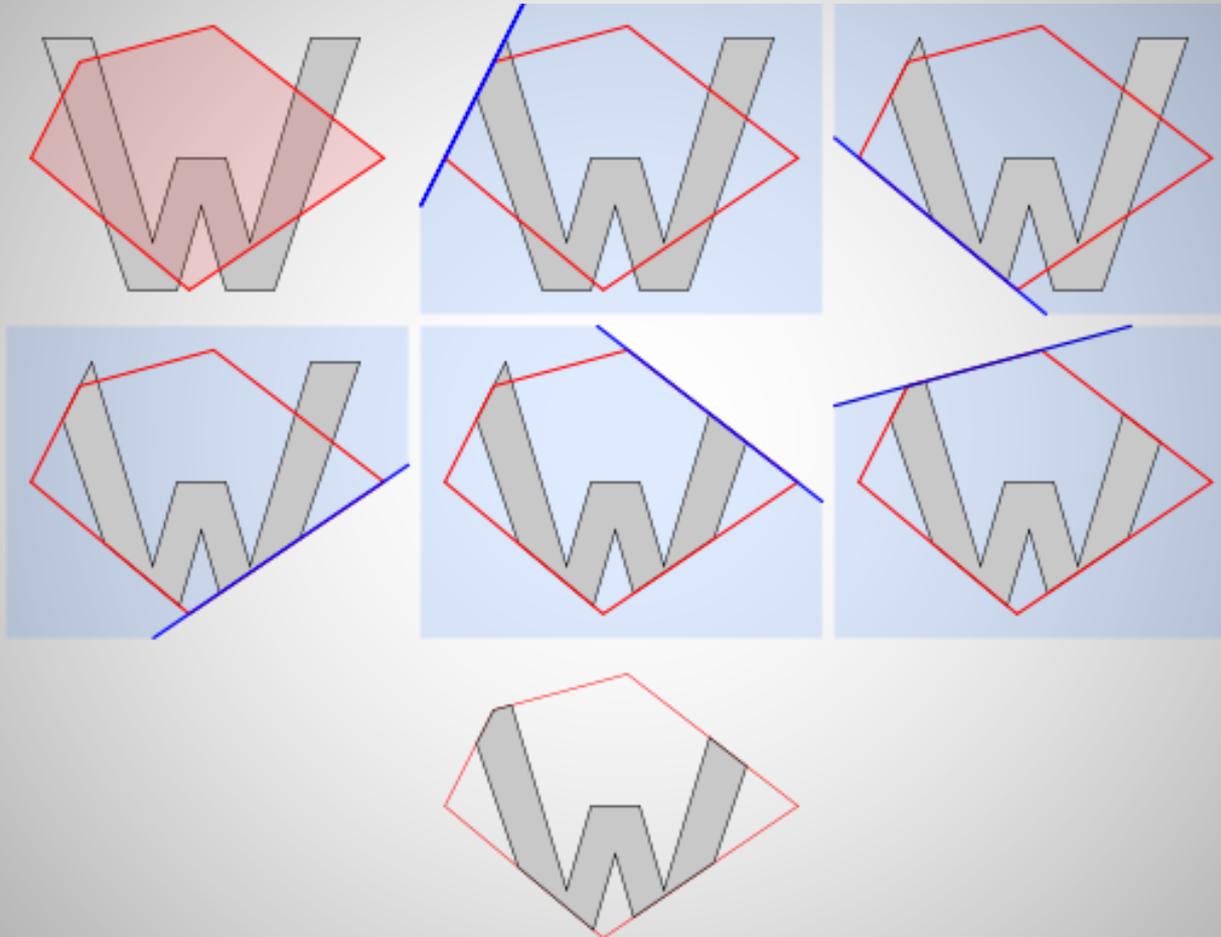


## Weiler-Atherton Algorithmus





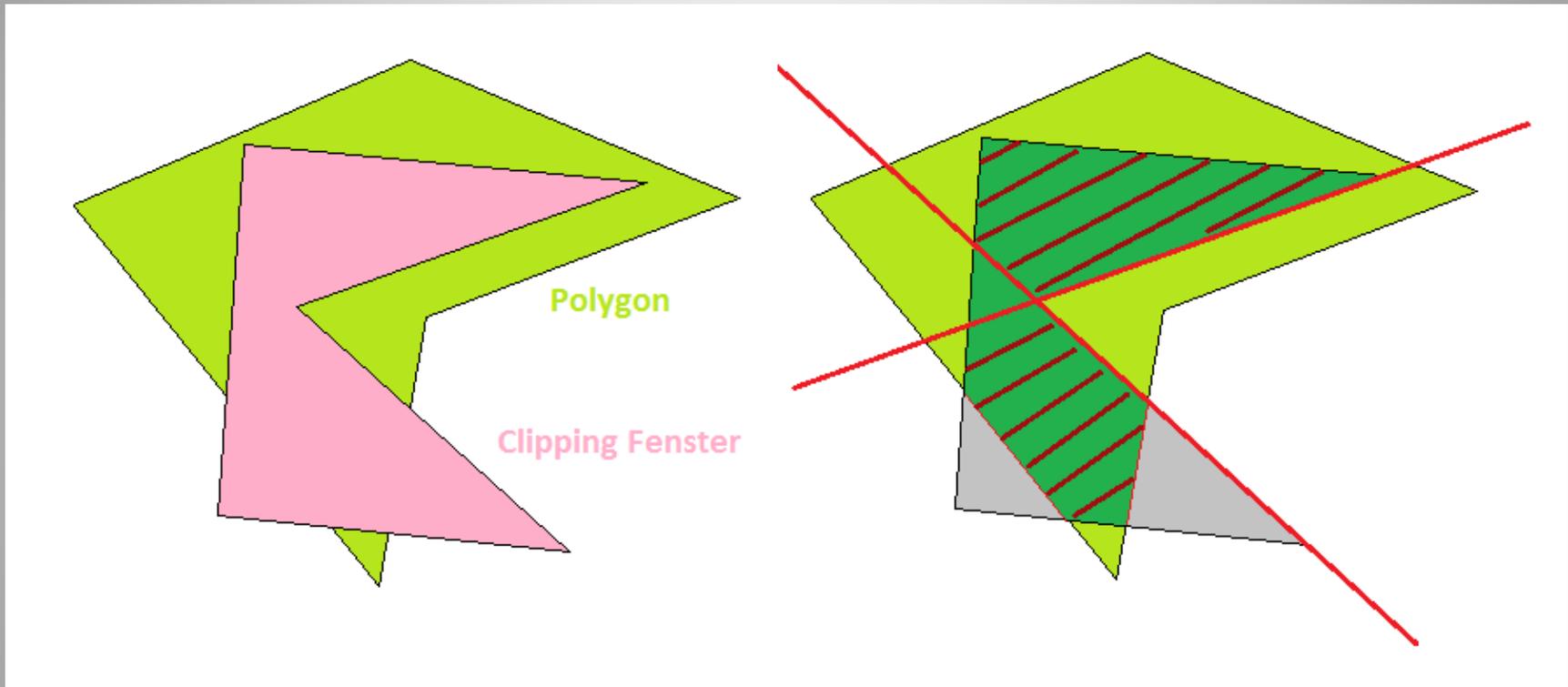
# Polygonclipping



- Nur konvexe Polygone möglich



# Polygonclipping



- valide Flächen werden weggeclipped



# Nutzen von Clipping

- Grafikspeicher wird nicht zugemüllt
- Halbkugeln zeichnen (abgeschnittene Objekte)





# Polygonoffsetting

**Problem:**

- Schwierig zwei sich schneidende Polygone zu Zeichnen.
- Führt zu Anzeigefehlern.

**Lösung:**

- Eins der Polygone kriegt ein Offset.
- Polygon wird an derselben Stelle vor einem anderen gezeichnet.



# Umlaufzahlen

- Windungszahlen
- **Definition:** Die Windungszahl einer Kurve  $\Gamma$  in Bezug auf einen Punkt  $z_0$  stellt die Anzahl der Umrundungen entgegen der Uhrzeigerrichtung um  $z_0$  dar, wenn man dem Verlauf der Kurve folgt. Eine Umrundung in Uhrzeigerrichtung ergibt eine negative Windungszahl



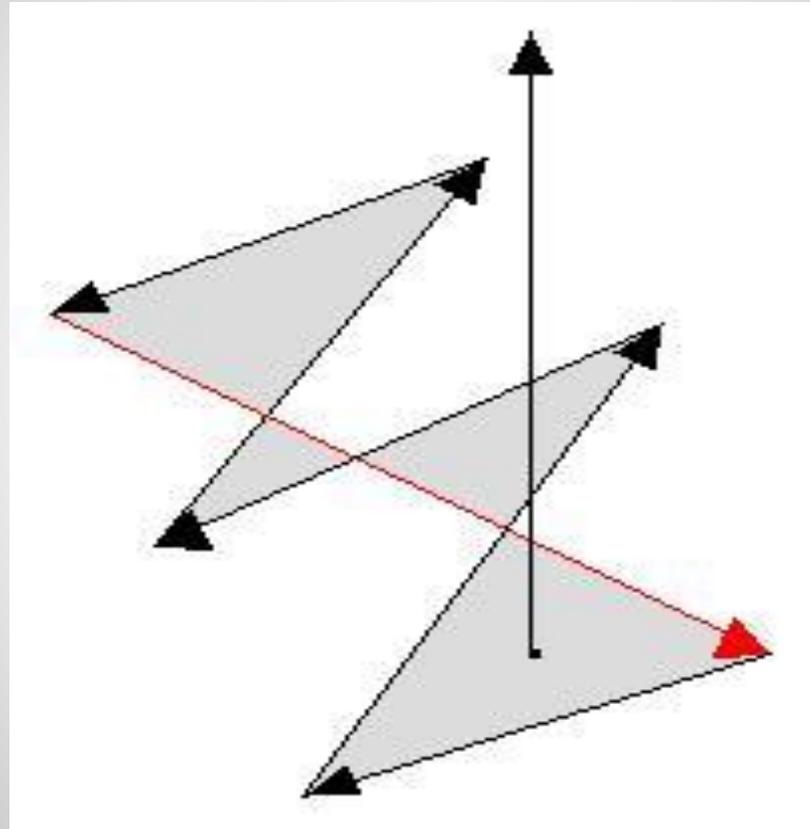
# Windungszahlen

Windungszahl = 1	Windungszahl = -1	Windungszahl = 0	Windungszahl = 1	Windungszahl = 2

Windungszahl = 0  $\Rightarrow$  außerhalb des Polygons

Windungszahl  $\neq$  0  $\Rightarrow$  innerhalb des Polygons

# Windungszahlen





# Windungszahlen

Schritt 1: Halbgerade zeichnen.

Schritt 2: Setze  $W = 0$ .

Schritt 3: Schnittpunkt mit Polygonkanten.

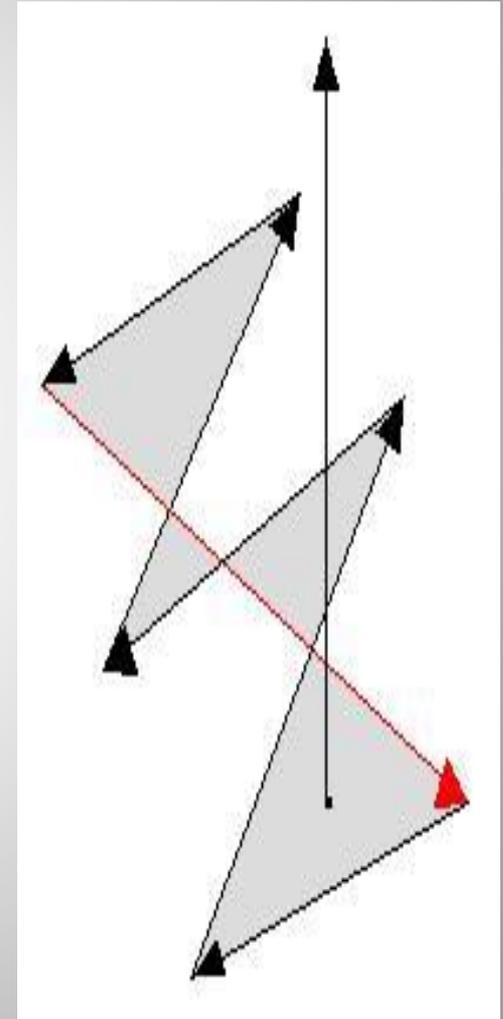
- Polygonkante verläuft von rechts nach links  $\Rightarrow W + 1$ .
- Polygonkante verläuft von links nach rechts  $\Rightarrow W - 1$ .

Schritt 4: Wiederholen für alle Schnittpunkte.

Schritt 5:  $W = \text{Windungszahl}$

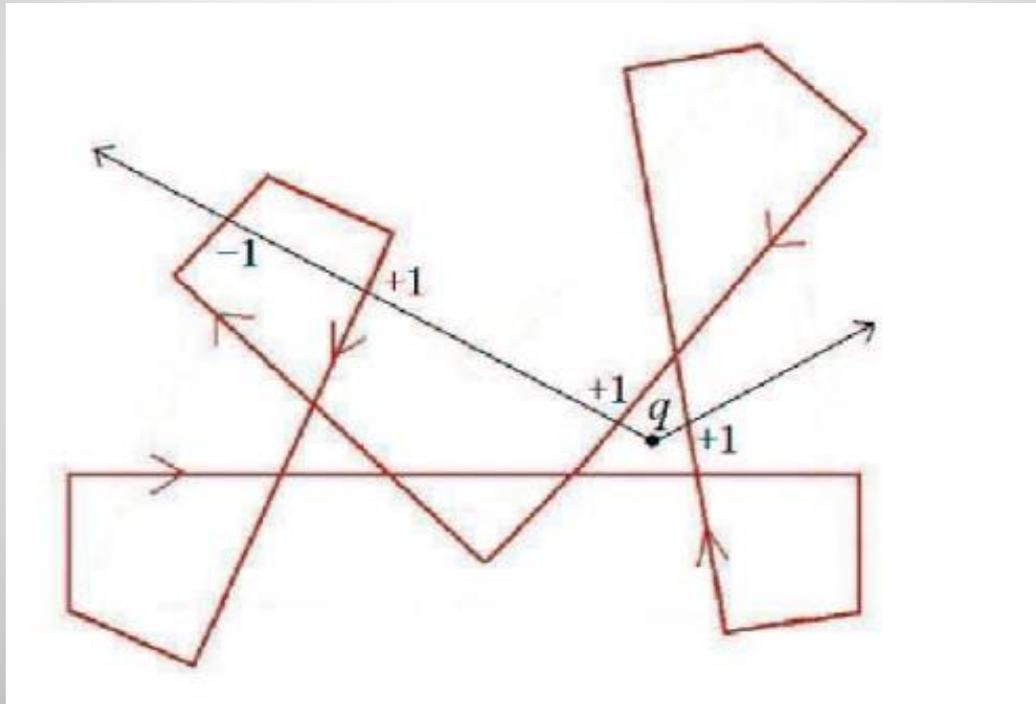
Windungszahl = 0  $\Rightarrow$  außerhalb des Polygons

Windungszahl  $\neq 0$   $\Rightarrow$  innerhalb des Polygons



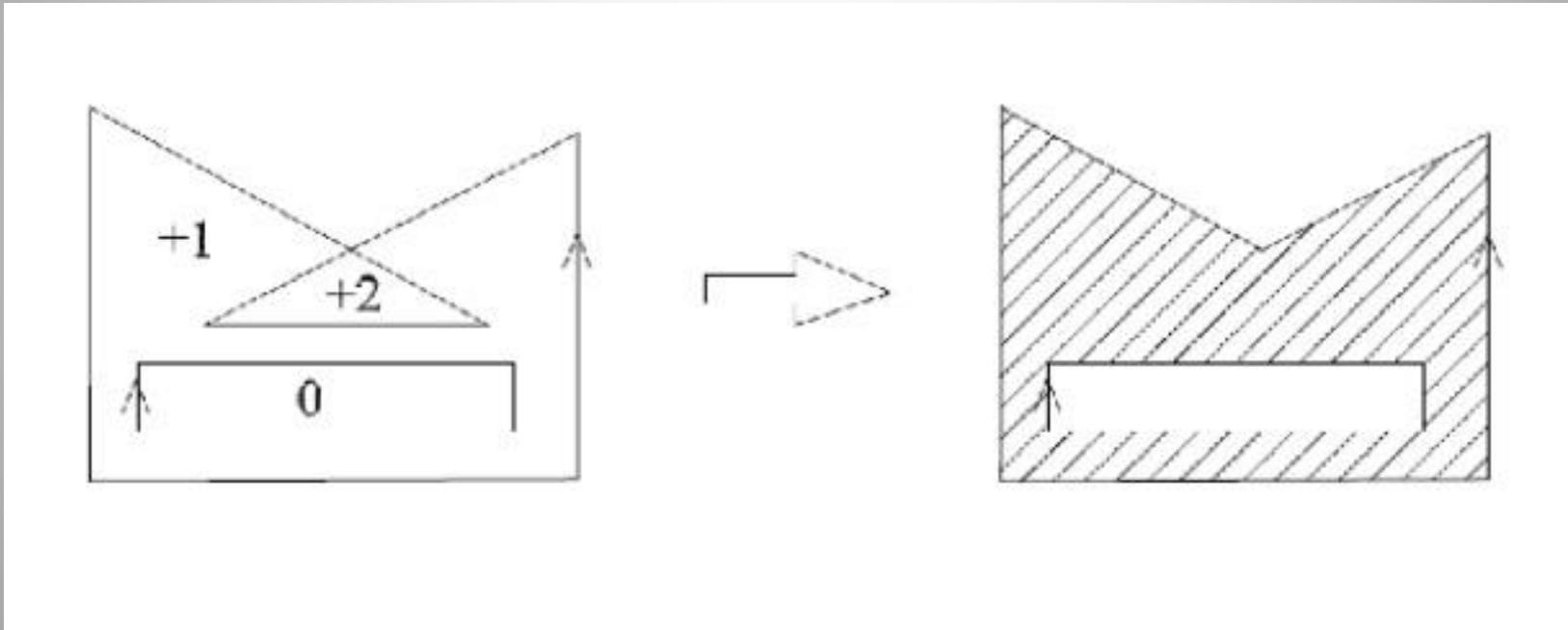


# Beispiel





# Beispiel





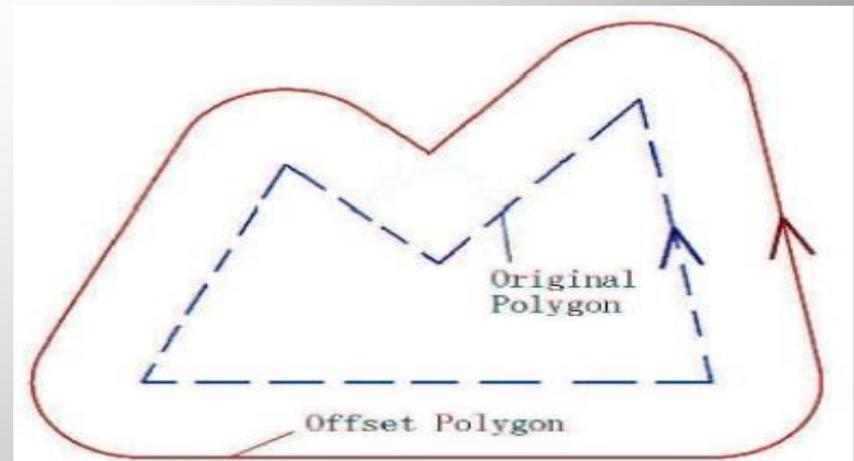
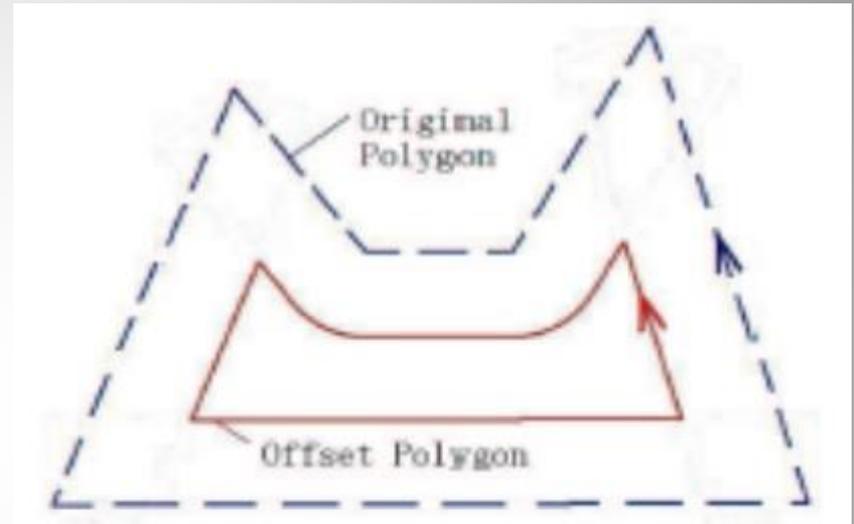
# Inneres / Äußeres Offset

## Inneres Offset

- Polygon wird nach Innen abgesetzt

## Äußeres Offset

- Polygon wird nach außen abgesetzt





# Inneres Offset

## Ablauf für inneres Offset

**Schritt 1:** Jede Kante nach innen absetzen

**Schritt 2:** Verbinden der Linien

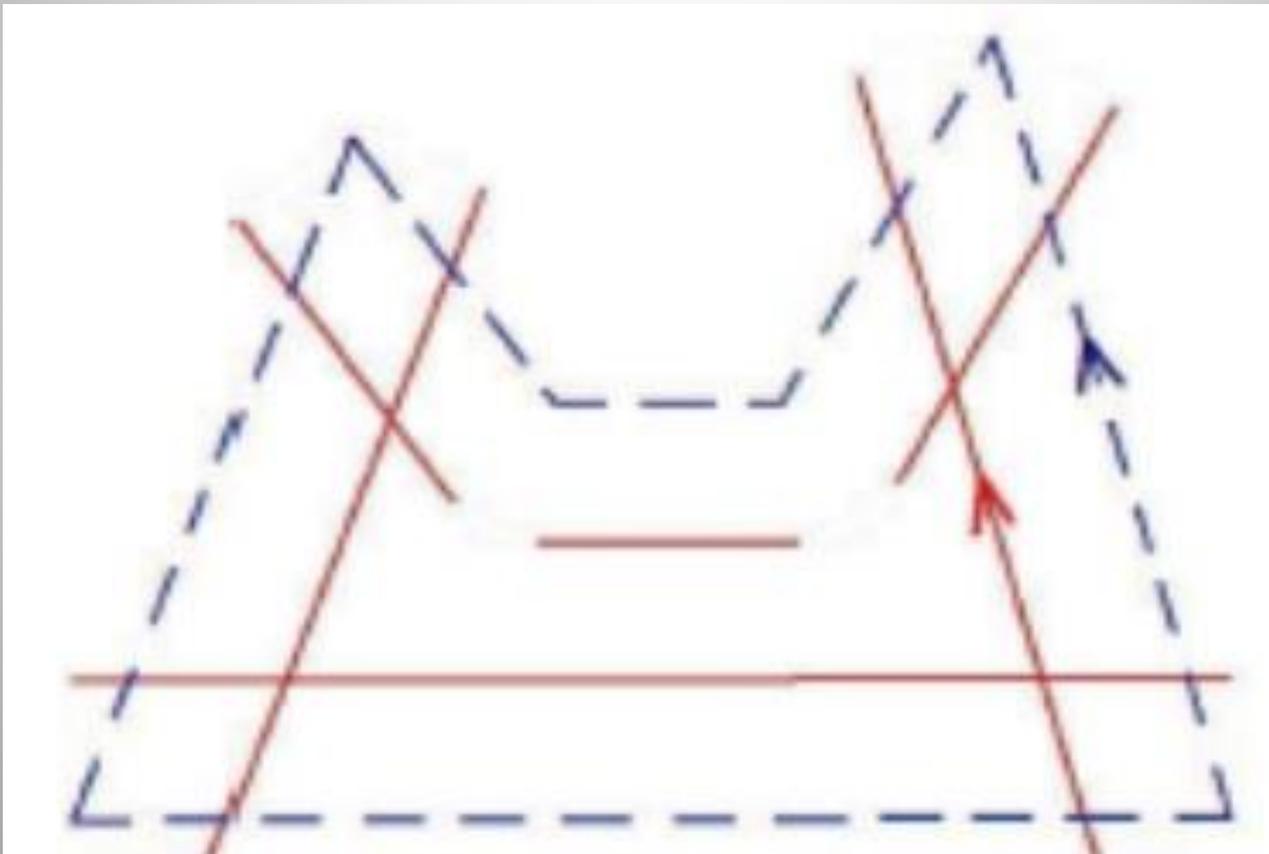
**Schritt 3:** Berechnung der Umlaufzahlen

**Schritt 4:** Entfernen von ungültigen Stellen



# Inneres Offset

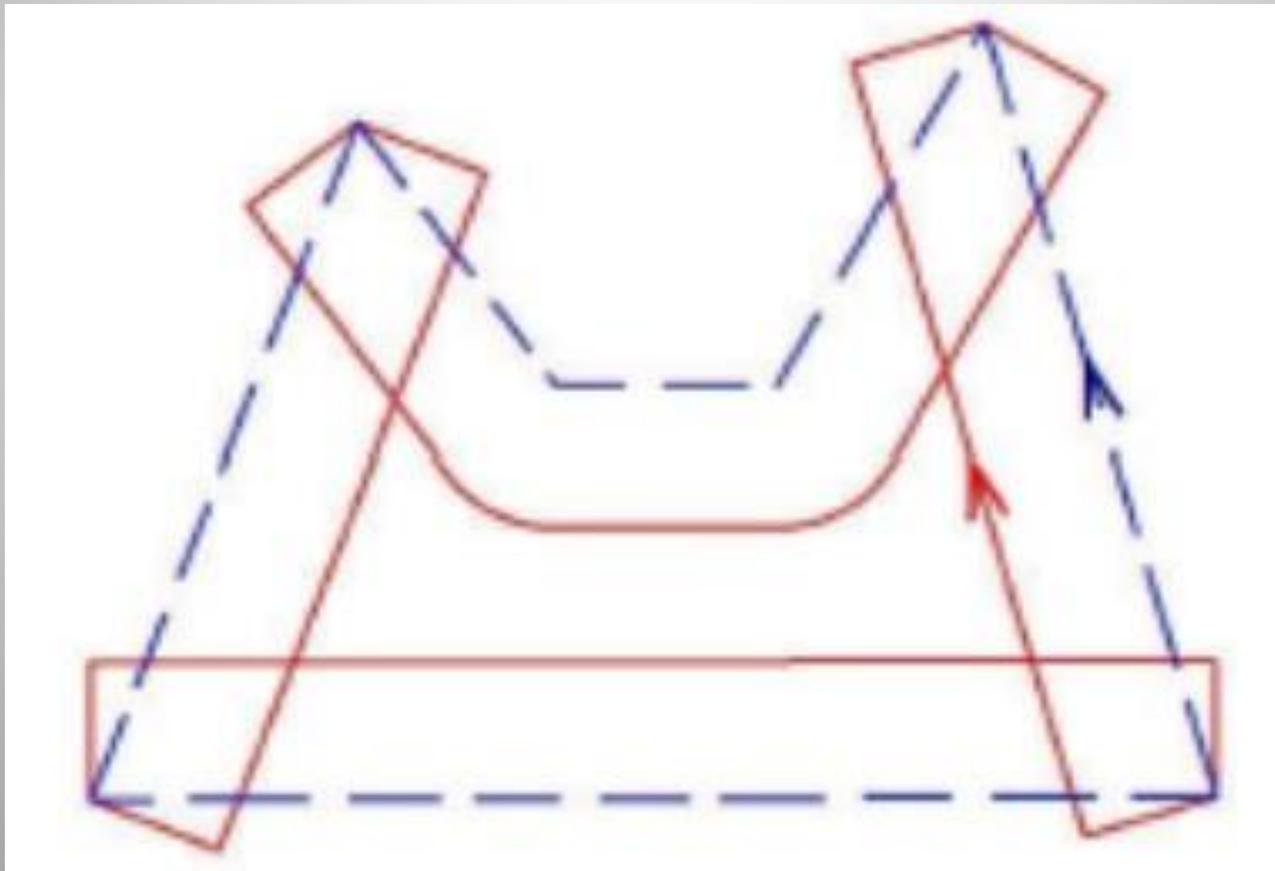
**Schritt 1:** Jede Kante nach innen absetzen





# Inneres Offset

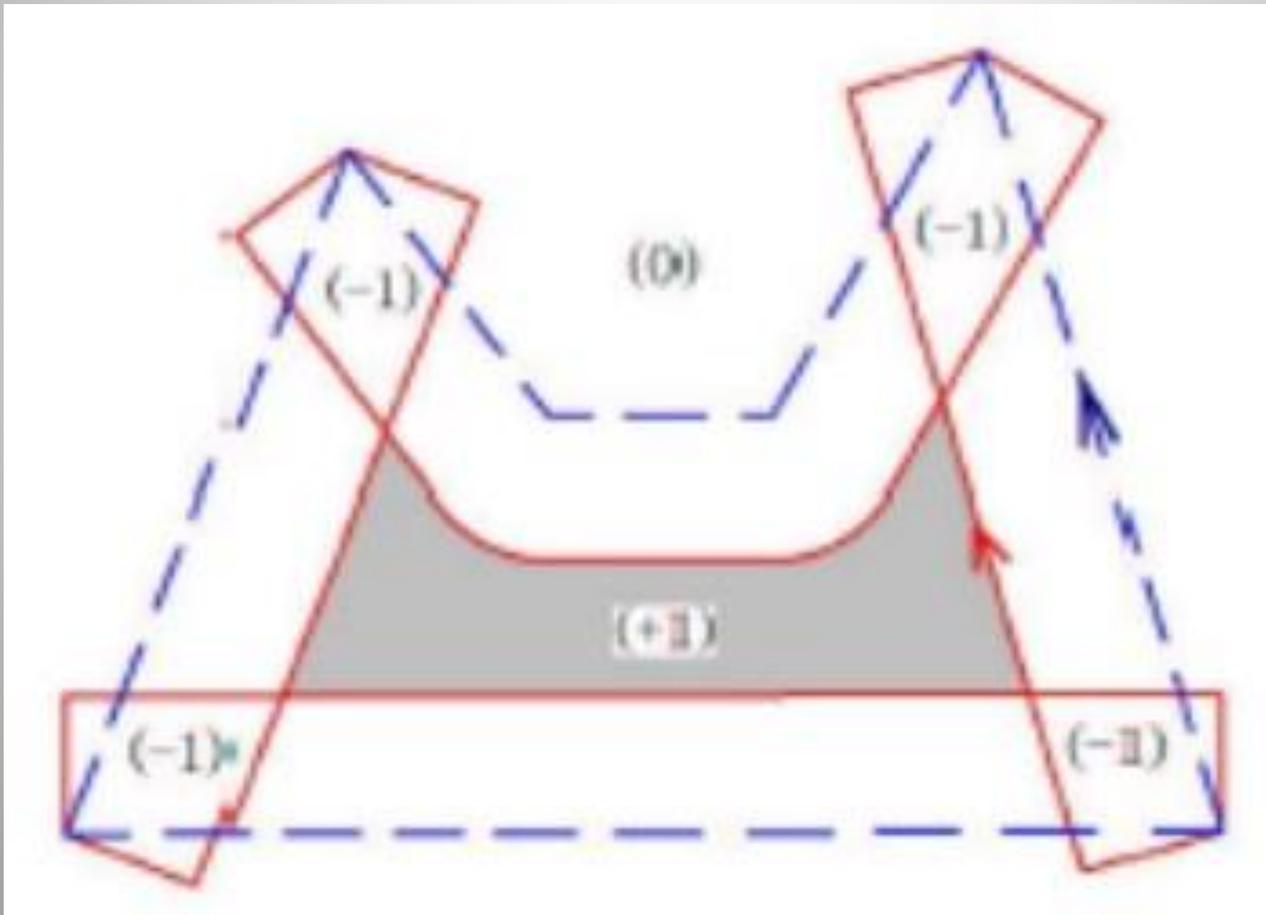
**Schritt 2:** Verbinden der Linien





# Inneres Offset

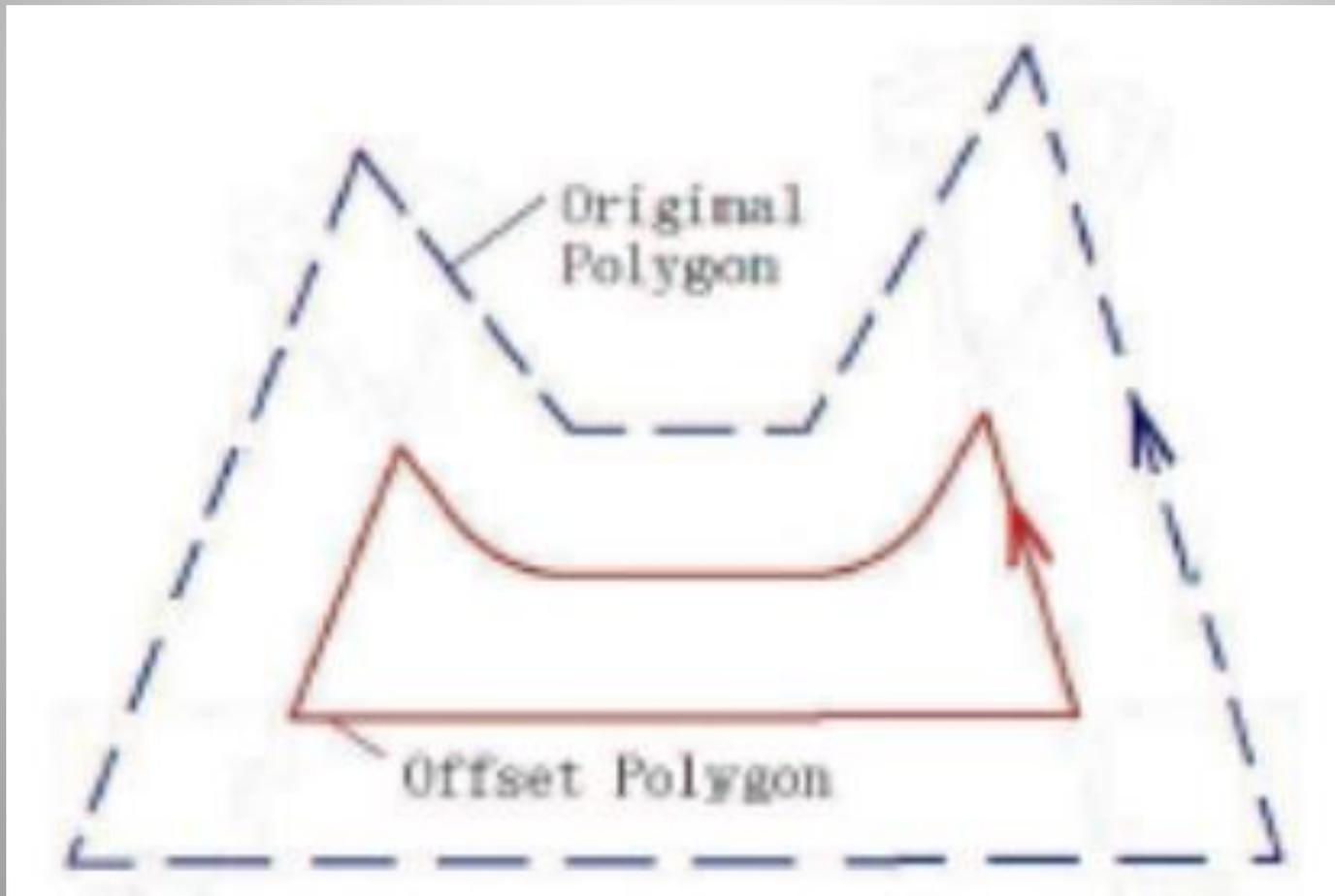
## Schritt 3: Berechnung der Umlaufzahlen





# Inneres Offset

**Schritt 4:** Entfernen von ungültigen Stellen





# Äußeres Offset

## Ablauf für äußeren Offsets

**Schritt 1:** Jede Kante nach außen absetzen

**Schritt 2:** Verbinden der Linien durch runde Bögen.

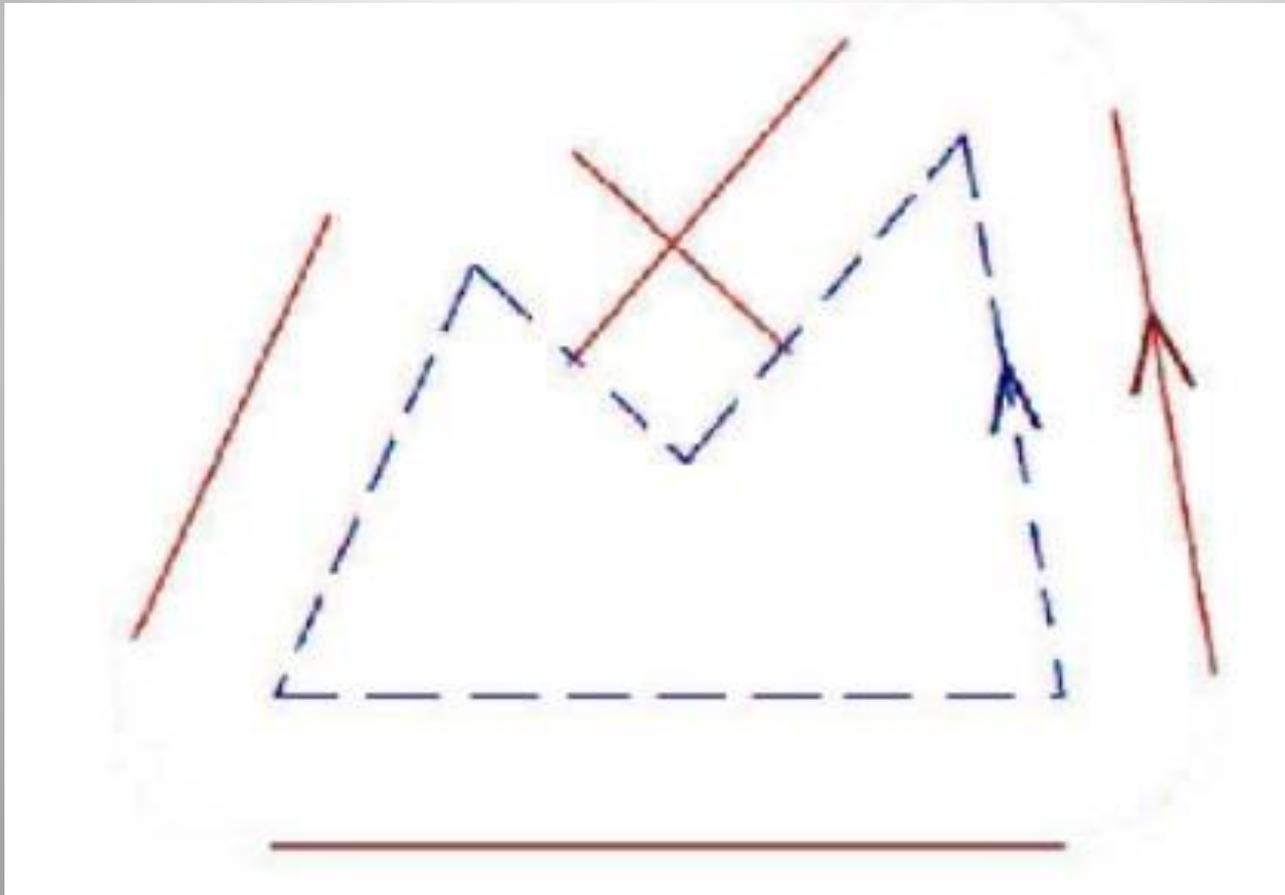
**Schritt 3:** Berechnung der Umlaufzahlen

**Schritt 4:** Entfernen von ungültigen Stellen



# Äußeres Offset

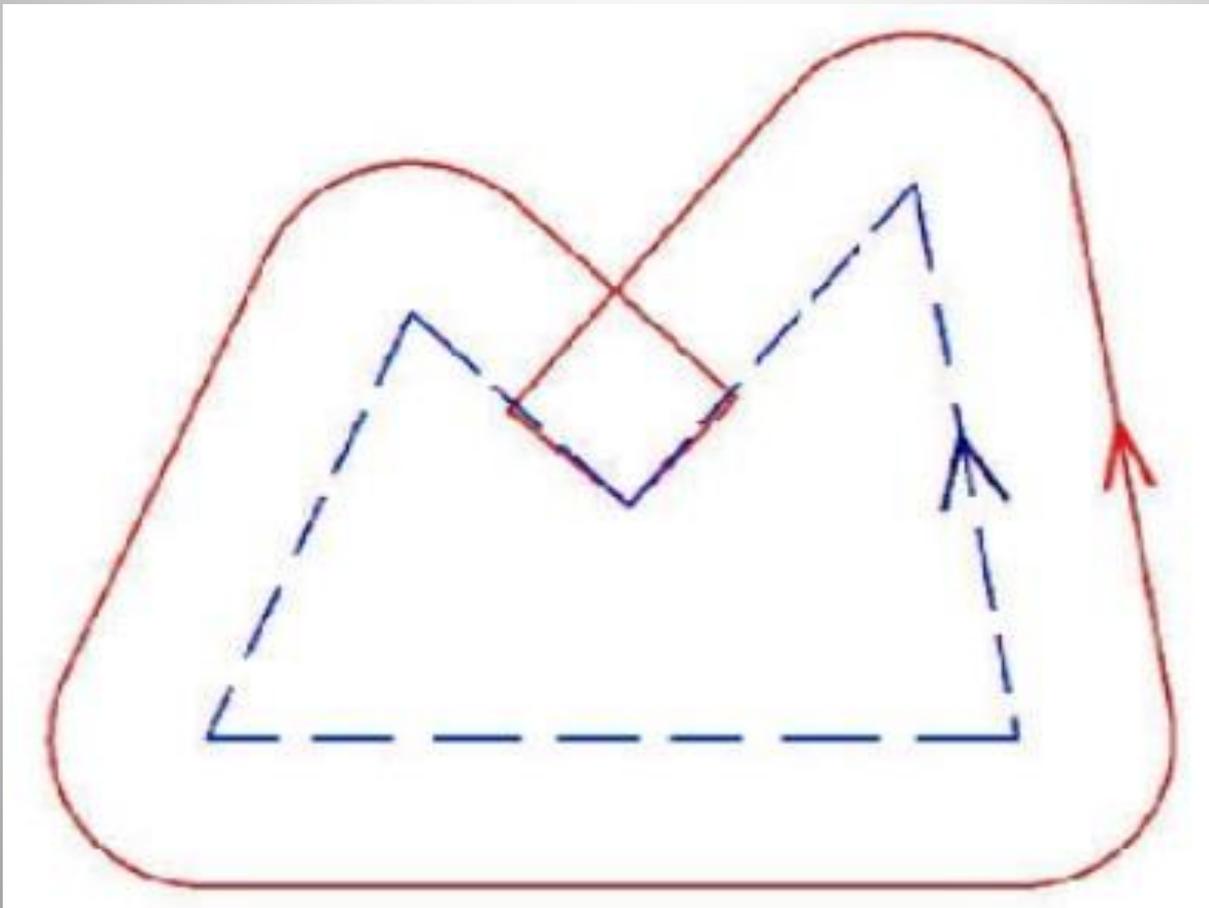
**Schritt 1:** Jede Kante nach außen absetzen



# Äußeres Offset



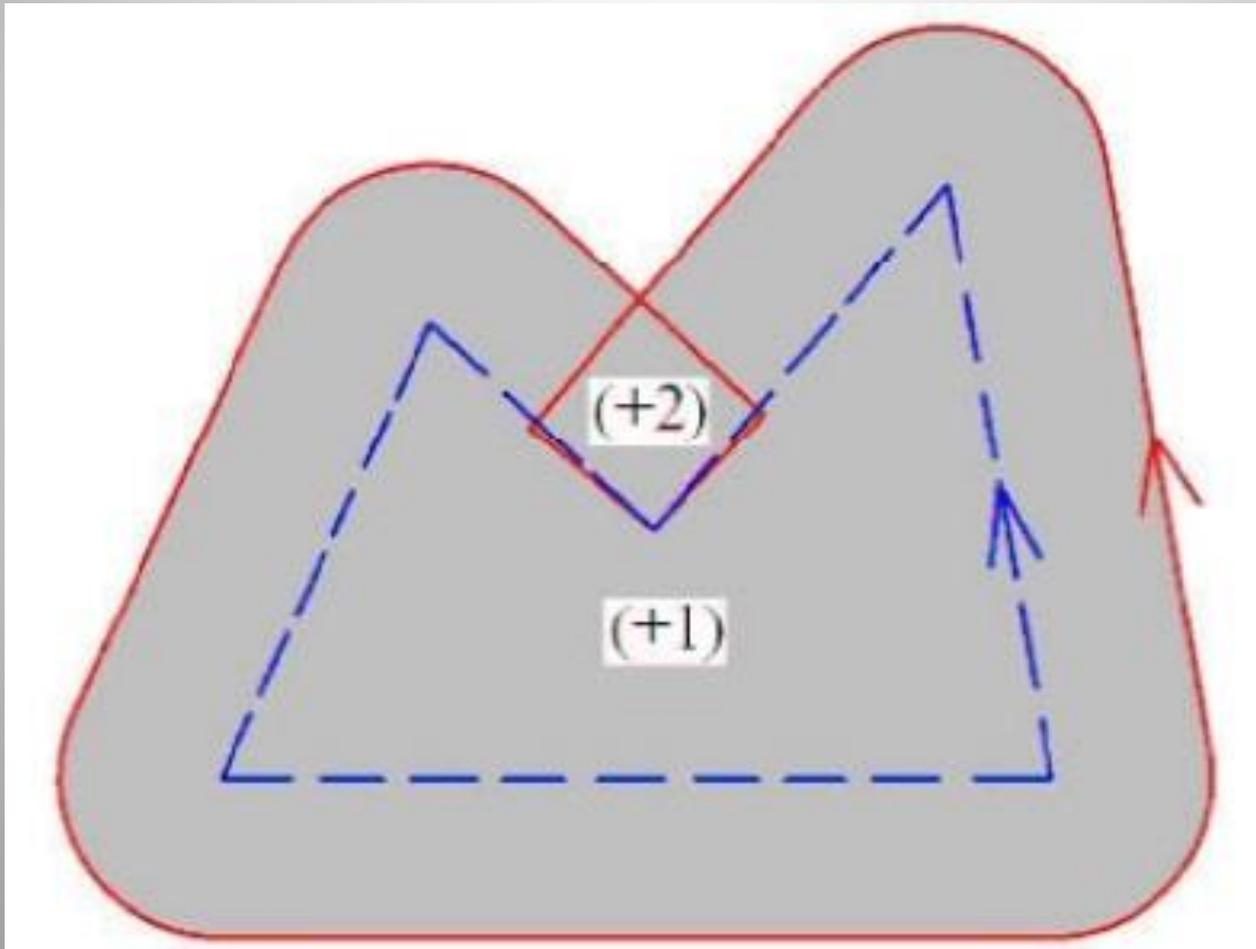
**Schritt 2:** Verbinden der Linien durch runde Bögen.





# Äußeres Offset

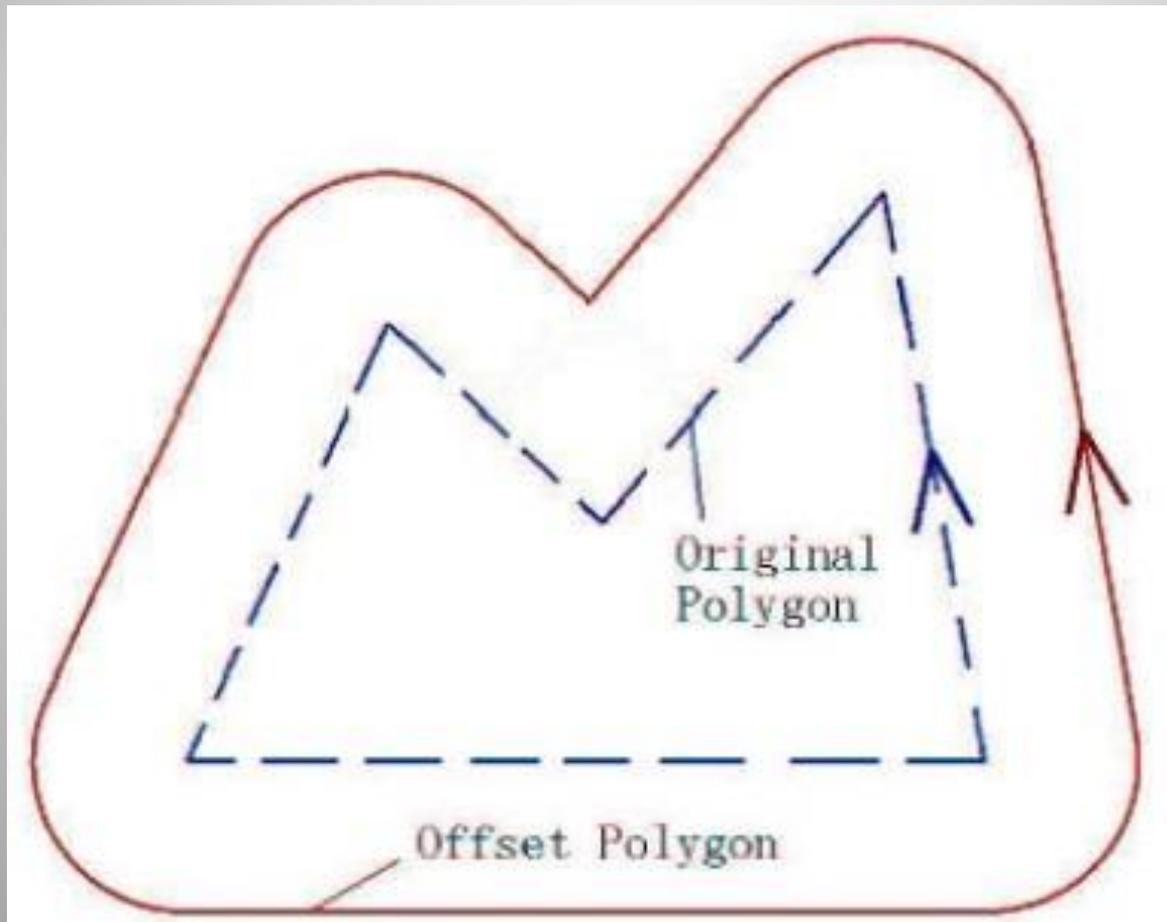
## Schritt 3: Berechnung der Umlaufzahlen



# Äußeres Offset

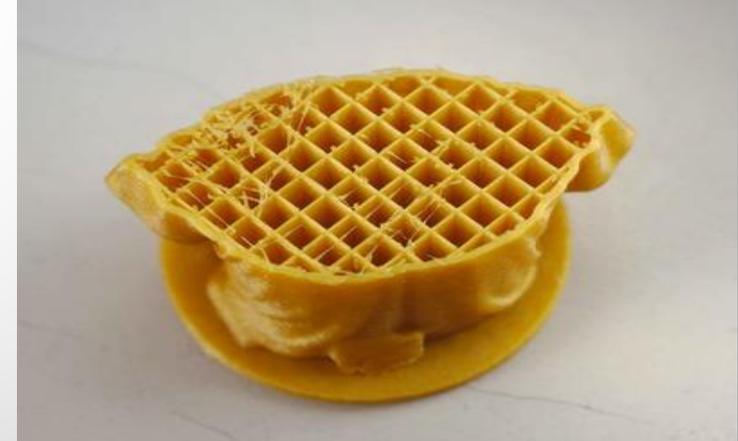
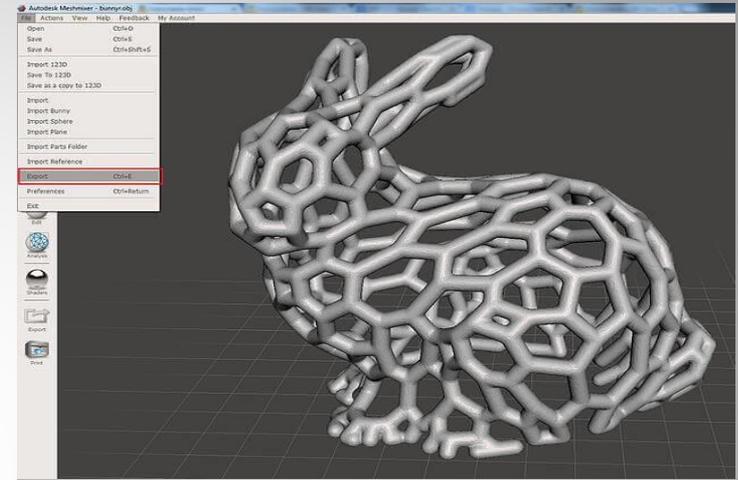
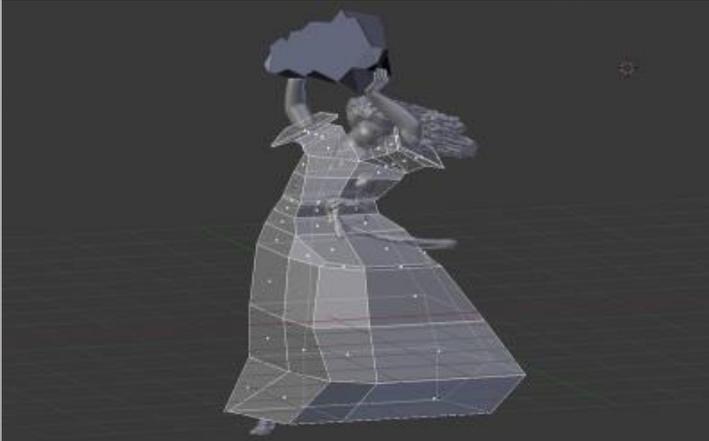
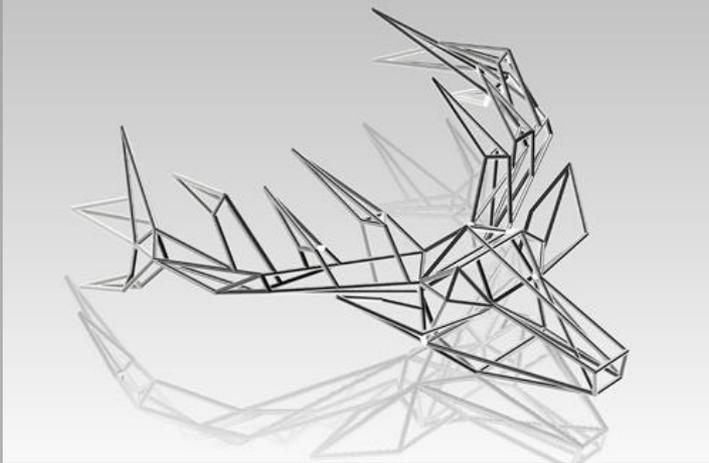


## Schritt 4: Entfernen von ungültigen Stellen





# Wozu braucht man Polygone im 3D Druck?



Jedes Objekt besteht aus einer Vielzahl von Polygonen!



# Quellen

- [https://de.wikipedia.org/wiki/Algorithmus\\_von\\_Cohen-Sutherland](https://de.wikipedia.org/wiki/Algorithmus_von_Cohen-Sutherland)
- [https://de.wikipedia.org/wiki/Clipping\\_\(Computergrafik\)](https://de.wikipedia.org/wiki/Clipping_(Computergrafik))
- [https://www.die-informatiker.net/topic/Computergrafik\\_SS11/Warum\\_Clipping/14861](https://www.die-informatiker.net/topic/Computergrafik_SS11/Warum_Clipping/14861)
- <https://wiki.delphigl.com/index.php/Clipping>
- <http://search.r-project.org/library/polyclip/html/polyoffset.html>
- [https://tams.informatik.uni-hamburg.de/lectures/2015ss/seminar/3dp/presentations/polygone\\_stenzel.pdf](https://tams.informatik.uni-hamburg.de/lectures/2015ss/seminar/3dp/presentations/polygone_stenzel.pdf)



# Quellen

- [http://delivery.acm.org/10.1145/130000/129906/p56-vatti.pdf?ip=134.100.29.211&id=129906&acc=ACTIVE%20SERVICE&key=2BA2C432AB83DA15%2EBB626F2563133BE7%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=793561939&CFTOKEN=53977179&acm=1464782871\\_50df05f11a6622bf7e25b016a639d525](http://delivery.acm.org/10.1145/130000/129906/p56-vatti.pdf?ip=134.100.29.211&id=129906&acc=ACTIVE%20SERVICE&key=2BA2C432AB83DA15%2EBB626F2563133BE7%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=793561939&CFTOKEN=53977179&acm=1464782871_50df05f11a6622bf7e25b016a639d525)
- <https://de.wikipedia.org/wiki/Windungszahl>
- <https://people.eecs.berkeley.edu/~sequin/CS285/PAPERS/OffsetPolygon.pdf>
- <http://www.3dsource.de/faq/polygonoffset.html>
- [https://www.researchgate.net/publication/256504043 Variable fused deposition modelling - concept design and tool path generation](https://www.researchgate.net/publication/256504043_Variable_fused_deposition_modelling_-_concept_design_and_tool_path_generation)



# Quellen

- Bilder:
- <https://mpi.fs.tum.de/wordpress/wp-content/uploads/2015/06/infill.jpg>
- <https://www.3d-grenzenlos.de/wp/wp-content/uploads/2015/05/3d-modell-voronoi.jpg>
- [https://www.rapidobject.com/csdata/image/1/de/3d-figur-modell-aus-polygonen-ergeben-ein-drahtgitternetz-polygonnetz\\_1024\\_400.jpg](https://www.rapidobject.com/csdata/image/1/de/3d-figur-modell-aus-polygonen-ergeben-ein-drahtgitternetz-polygonnetz_1024_400.jpg)
- [http://ulrichmerz.com/wordpress/wp-content/gallery/jgm/jgm\\_08\\_web\\_big.jpg](http://ulrichmerz.com/wordpress/wp-content/gallery/jgm/jgm_08_web_big.jpg)
- Bilder aus dem Paper
- Selbstgemachte Bilder