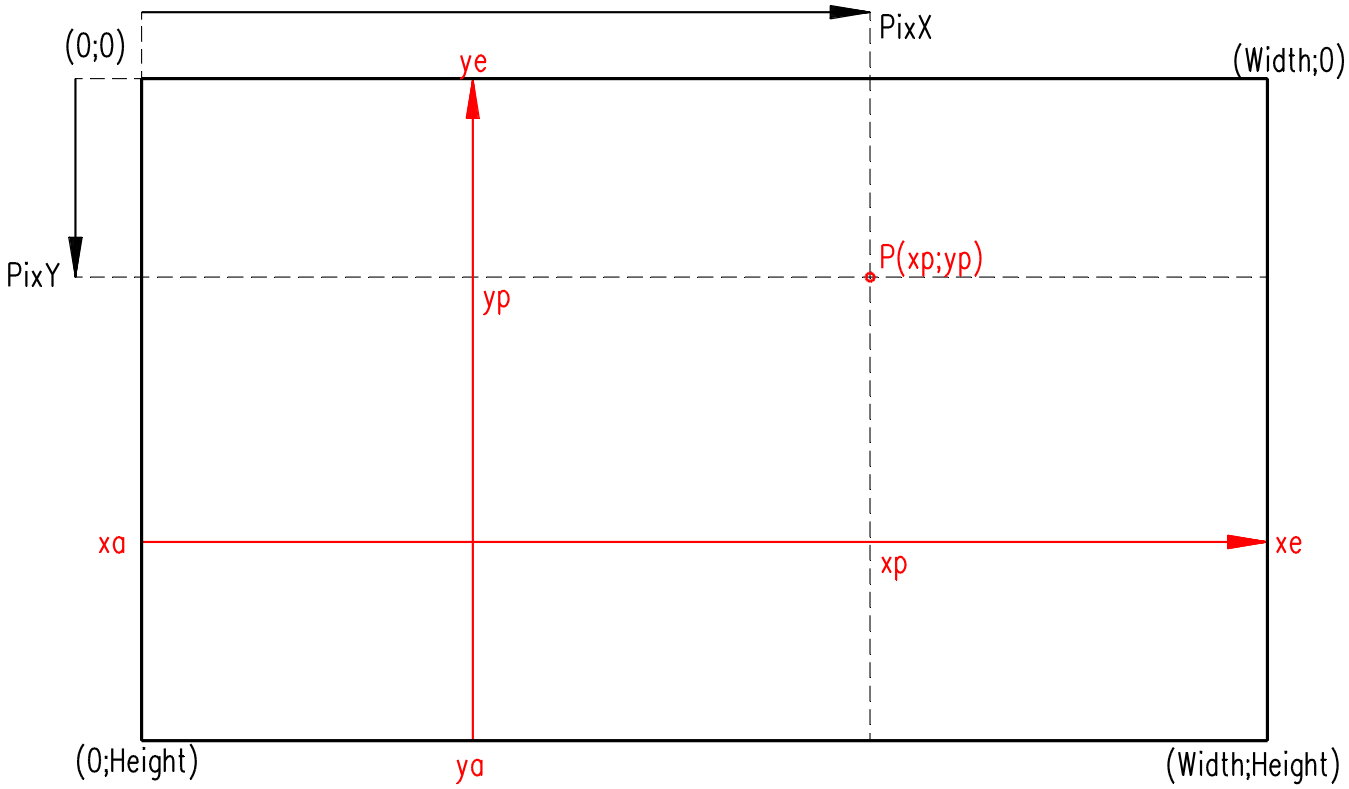
2D-Koordinatensysteme bei Computern Ac

Die Computergrafik kennt nur natürliche Zahlen als Pixelkoordinaten, im folgenden PixX und PixY genannt . Ein Pixel hat dann die Koordinaten **Pix (PixX ; PixY)** .
Der x-Bereich läuft von **0 bis Width** (von links nach rechts) .
Der y-Bereich läuft von **0 bis Height** (von oben nach unten !) .
Will man nun einen beliebigen Punkt **P (xp ; yp)** mit reellen Koordinaten xp, yp zeichnen, so muss man diese in Pixelkoordinaten umrechnen . Die Grafik zeigt den Sachverhalt :



Auf Grund der Strahlensätze gelten für xp≠xa und yp≠ye :

$$\frac{PixX}{xp-xa}=\frac{Width}{xe-xa} \quad \text{sowie} \quad \frac{PixY}{ye-yp}=\frac{Height}{ye-ya} \quad .$$

Wegen der Ganzzahligkeit der Pixelkoordinaten folgt mittels Round :

$$PixX(xp)=Round\left(\frac{xp-xa}{xe-xa}\cdot Width\right) \quad sowie \quad PixY(yp)=Round\left(\frac{ye-yp}{ye-ya}\cdot Height\right)$$

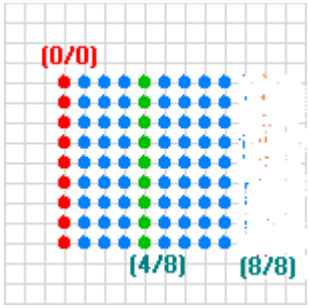
Entsprechend lassen sich die Gleichungen nach xp, yp auflösen :

$$xp(PixX)=xa+\frac{PixX\cdot(xe-xa)}{Width} \quad sowie \quad yp(PixY)=ye-\frac{PixY\cdot(ye-ya)}{Height}$$

Beispiel: xa=-2,5 xe=7,5 ya=-5 ye=3,5 sei als „User-KOS“ vorgegeben.
Außerdem habe der Bildschirm eine Auflösung von Width x Height = 1680 x 1050 .
Dann hat der Punkt P(3,1/-1,8) die (ganzzahligen) Pixelkoordinaten (PixX/PixY)=(941/655).

Achtung: Im Programmiersystem **Delphi** sind mit „width“ und „height“ nicht Breite und Höhe gemeint, sondern die entsprechende **Anzahl der vorhandenen Pixels**. Entsprechend heißen diese Variablen in der Programmiersprache **Java** getWidth() bzw. getHeight() .

Den Unterschied verdeutlicht das **Beispiel** eines „Mini-Bildschirms“ mit Breite = 8 und Höhe = 8



Man sieht, dass es in Breite und Höhe jeweils 9 (!) Pixels gibt, also jeweils 1 Pixel mehr als die festgelegte Breite und Höhe des Bildschirms. Das Pixel Nummer 0 darf also bei der Breiten- und Höhenberechnung nicht mitgezählt werden. Das in der Mitte befindliche grüne Pixel (4/8) steht an der 5. Stelle, während das am Ende befindliche Pixel (8/8) an der 9. Stelle steht .

Für die Umrechnungsformel gilt daher, dass im Programmiersystem **Delphi**
Width durch (Width-1) und Height durch (Height-1) ersetzt werden müssen .
Ebenso muss man in **Java** getWidth() durch getWidth()-1 und getHeight() durch getHeight()-1 ersetzen !

Umwandlung zwischen Pixelkoordinaten und Userkoordinaten in DELPHI

```
function xPix (image:TImage;xUser:extended):integer;
begin Result:=round((xUser-xa)/(xe-xa)*(image.width-1)); end;

function yPix (image:TImage;yUser:extended):integer;
begin Result:=round((ye-yUser)/(ye-ya)*(image.height-1)); end;

function xUser (image:TImage;xPix:integer):extended;
begin Result := xa+xPix*(xe-xa)/(image.width-1); end;

function yUser (image:TImage;yPix:integer):extended;
begin Result := ye-yPix*(ye-ya)/(image.height-1); end;
```

Umwandlung zwischen Pixelkoordinaten und Userkoordinaten in JAVA

```
public int xPix(double xUser) { // mit math. exakter Rundung
     return (int) ((xUser - xa) / (xe - xa) * (getWidth-1) + .5);
}

public int yPix(double yUser) {
     return (int) ((ye - yUser) / (ye - ya) * (getHeight-1) + .5);
}

public double xUser(int xPix) {
     return xa + xPix * (xe - xa) / (getWidth-1);
}

public double yUser(int yPix) {
     return ye - yPix * (ye - ya) / (getHeight-1);
}
```